**Lab 03 – Objects and methods**

**COMP130 - Introduction to Computing**

**Dickinson College**

**Instructor: John MacCormick**

There is no responses document for this lab. Submit your Python scripts to Moodle.

**Don’t comment out earlier attempts or questions as in previous labs. Submit just the final working version of the code in each script.**

As described in class, we use John Zelle’s graphics.py module for this lab. Links to the documentation and code for this module are on the course homepage. Simple example programs using the graphics module were explained in class and are available on the course “detailed schedule and resources” page. Please also see the supplementary study guide for an explanation of how to set the correct coordinate system.

**We do not use the turtle module in this lab. Please use graphics.py.**

**Qu 1.** Write a program called shapes.py that draws 4 different geometric shapes of 4 different colors, each near a corner of the graphics window. No shape should touch the window's edges. Add a light blue diamond in the middle of the window. As an optional extra challenge, ensure that the diamond is exactly centered.

A screenshot of a computer screen

Description automatically generated**Qu 2.** Write a program called checkers.py that shows a standard checkerboard centered within a window, similar to the one shown here. The distance between the edge of the board and the edge of the window must be the same as the width of a square. The checkerboard is not required to use multiple colors. For a small amount of extra credit, give the squares two different colors in the alternating pattern shown here.

Hints:

a) Before you start coding, identify the smallest unit you must use to build up this image.

b) Create a variable to be that unit.

c) Use that variable to create everything else, including the graphics window.

d) First create a single square, then a row of squares (use a simple loop), then a grid of squares (use a nested loop).

A screenshot of a castle

Description automatically generated**Optional** **Qu 3.**

Write a program called castle.py that shows... a castle. The castle could be built of offset rectangular units (like brickwork) and have walls, towers, battlements and an entry of some type. The picture shown here may be a useful starting point, but feel free to be creative. To really impress, use the color\_rgb( ) function to make the rectangular units change color in a cool way.

**Acknowledgment.** This lab was originally authored by Lev Fruchter. It was adapted and edited by John MacCormick.