CSIT 104 – Computational Concepts

Python HW 4 (40 pts)

PART 1 (20 pts)
Write a function titled drawPolygon() that has one parameter (the number of sides for the polygon). Using the value received as parameter, draws a regular polygon using “for”

PART 2 (20 pts)
Write a function titled drawPolygonFilled() that has three parameters (the number of sides for the polygon, the color of the edge, and the color of the fill). Using the value received as parameter, draws a regular polygon using “for”, colors the edges and fills the shape according to the parameter values. The parameters for the colors must be in hex notation.

Helper Documentation – The Problem

You will use Turtle graphics

1. Loops: To draw a regular polygon given the number of sides provided by the user.
2. Selection:
    a. Add the capability to your program to fill polygons with a color indicated by a combination of the colors red, green and blue provided by the user.
    b. You will check to ensure that correct color values are input.

Originally written as a part of the *logo* programming language, Turtle graphics (http://en.wikipedia.org/wiki/Turtle_graphics) is one of the oldest graphics programs. It is a 2D graphics package that uses a Cartesian coordinate system and a "turtle," which you can imagine has a pen attached to its body. The turtle can move around the plane, drawing as it goes. Python has a module that implements the behavior of the original turtle graphics program and this module is simply called "turtle"

Using turtle graphics:

In order to use turtle graphics in Python you must first import the turtle module. You can then use the help function in idle to find out what methods this module includes and what they do. Just type "import turtle" in the idle command window, hit enter, and then type help(turtle) and scroll up through the list and information. For more details Google "Python 3 turtle." A sample Python program "turtleSample.py" is provided.

The basic concept behind the turtle is the pen. The pen is either up or down. When down, the turtle draws as it moves around the Cartesian graph. There are a number of methods that are needed in this project:

`turtle.up(),turtle.down()`: Set the pen state to be up (not drawing) or down (drawing)

`turtle.right(degrees), turtle.left(degrees)`: Turn the direction that the turtle is facing. The amount of turn is indicated in degrees.

`turtle.forward(distance), turtle.backward(distance)`: Move the turtle forward or backward the amount of distance indicated. Depends on the direction the turtle is facing. Draws a line if the pen is down, not if the pen is up.

`turtle.goto(x,y)`: The goto method moves the turtle to a specified point, drawing a line along the way if the pen is down, and not drawing if the pen is up. Note: The turtle always starts at the point (0,0). The goto method moves to the indicated x,y coordinates.

`turtle.color(r,g,b)`: The color method sets the color that the pen will hold for all drawing until the color is changed. It takes three arguments, each a floating-point number between 0.0-1.0. The first is the amount of red, the second, the amount of green and the third the amount of blue.

`turtle.begin_fill(), turtle.end_fill()`: Use the command turtle.begin_fill() before you start drawing a figure. Draw the figure, then execute the command turtle.end_fill(). The figure drawn between the two fill commands will be filled with the present color setting.

`turtle.bye()`: Close the turtle drawing window.

First, try it!
The first thing you should do is try out some of the turtle commands by just typing them in the idle window. You can get a much better feel for how the whole thing works by just trying it.

Here is how to draw a line, turn left 45 degrees, and then draw another line:

```
>>> import turtle
>>> turtle.down()
>>> turtle.forward(20)
>>> turtle.left(45)
>>> turtle.forward(40)
>>> turtle.bye()
```

Regular Polygon
The formula for the total degrees of all angles on the inside of a regular polygon with *n* sides is *180 degrees * (n – 2)* so each interior angle is *180*(n – 2)/n.*
From that you can calculate how much to turn (exterior angle): *180 - 180*(n – 2)/n.*
Use a loop to draw the regular polygon.

Loops

"for" is useful when you know how many times you want to repeatedly do something, e.g. you know how many sides of a polygon you want to draw. Here is an example showing how to print something **n** times.

```
>>> n = 5
>>> for i in range(n):
     print("Hi!")


Hi!
Hi!
Hi!
Hi!
Hi!
```

"while" is most useful when you don't know when writing the program that you will be repeating a fixed number of times, e.g. asking for input until correct input is provided. Here we will use "while" for a fixed number of times simply to practice using "while." To contrast with "for" we will print something **n** times. Note how with the "while" we have to manage the counter variable "count."
If you mess up the count management, you may end up repeating forever. In that case use "Ctrl-c" to stop the loop—that is, hold down the key named "control" and then the "c" key while still holding down the "control" key.

```
>>> n = 5
>>> count = 0
>>> while count < n:
     print("Hi!")
     count = count + 1
```

```
Hi!
Hi!
Hi!
Hi!
Hi!
```

PART 2

Your program will read in color values from the user and then use the methods in the turtle module to draw the polygons (above) filled with the color indicated. There are many ways to create a color but a common one used in computer graphics is the process of additive color (see http://en.wikipedia.org/wiki/Additive_color). Combining different amounts of red, green and blue can create most, but not all, colors.

For example, here is how to draw a red circle: all red, no green, no blue.

```
>>> import turtle
>>> turtle.color(1,0,0)
>>> turtle.begin_fill()
>>> turtle.circle(20)
>>> turtle.end_fill()
>>> turtle.bye()
```