Lesson 02

CSC140 Foundations of Computer Science

13 February 2020

```
# A program that draws cardoid curves.
# Margaret Hamilton
# CSC140 Foundations of Computer Science
# 13 February 2020

import math
import turtle

# A larger number of steps will give
# a smoother curve.
# TO-DO: Experiment with smaler and larger values for STEPS.
STEPS = 64

# Specify the width and height of the window.
WIDTH = 512
HEIGHT = 512

def draw_curve( pen, scale_factor, rotation_angle ):
    # Put the pen at the origin.
    pen.up()
    pen.goto( 0, 0 )
    pen.down()

    # This loop takes us around a circle in equal-sized steps.
    for i in range(STEPS):
        # What fraction of the distance around the circle?
        fraction = i / STEPS
        # Compute the angle in radians.
        angle = fraction * 2 * math.pi

        # Here is the formula for a cardoid curve.
        # The formula is expressed in polar coordinates.
        # TO-DO: Experiment with formulas for other kinds of curves.
        r = scale_factor * (1 - math.cos( angle ))
```

```
            # Produce Cartesian coordinates from
            # the polar coordinates
            x = r * math.cos( angle )
            y = r * math.sin( angle )

            # Construct the rotation matrix.
            cosine = math.cos( rotation_angle )
            sine = math.sin( rotation_angle )

            # The matrix has 4 elements.
            m00 = cosine
            m01 = -sine
            m10 = sine
            m11 = cosine

            # Apply the rotation matrix.
            # (Rotate the point about the origin.)
            x_rotated = m00 * x + m01 * y
            y_rotated = m10 * x + m11 * y

            # Draw the next segment of the curve
            pen.goto( x_rotated, y_rotated )

        # end of draw_curve()

    if __name__ == '__main__':
        window = turtle.Screen()
        # TO-DO: Experiment with other background colors.
        window.bgcolor( "Sea Green" )
        window.screensize( WIDTH, HEIGHT )

        pen = turtle.Turtle()
        # TO-D0: Experiment with other pen widths.
        pen.width( 8 )
        # TO-DO: Experiment with other colors for the curve.
        pen.color( "goldenrod" )

        # A value of 128 for the parameter 'a' yields a
        # nice picture when the width and height of the
        # picture is 512.
        scale_factor = int( input( "Enter a positive integer value: " ) )

        # TO-DO: Experiment with different numbers of copies of the curve.
        number_of_copies = 3
        for i in range( number_of_copies ):
```

```
        rotation_angle = i * math.pi/number_of_copies
        draw_curve( pen, scale_factor, rotation_angle )

    # TO-DO: Try adding another loop. Maybe different sizes
    # or colors for these additional curves?

    # The program ends when the user moves the mouse to
    # put the cursor inside the picture and then clicks
    # the left button on the mouse.
    window.exitonclick()

    # end of main function
```