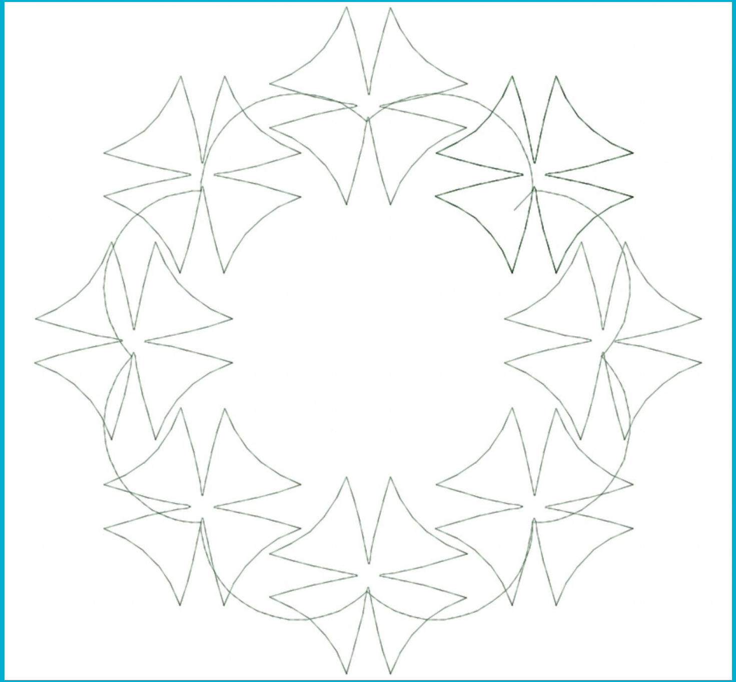
MECE4606 Digital Manufacturing Assignment 3 - Programmable Embroidery

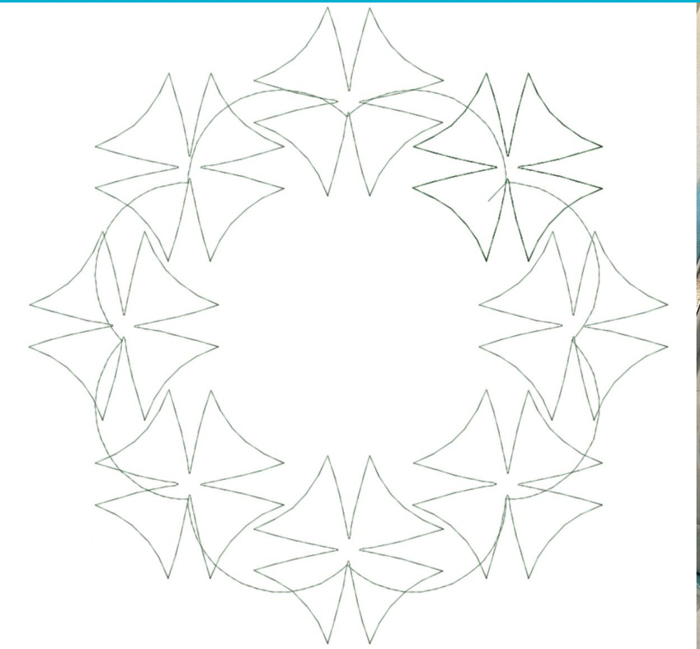
Hansen Ding (hd2521), Yibo Peng (yp2644)

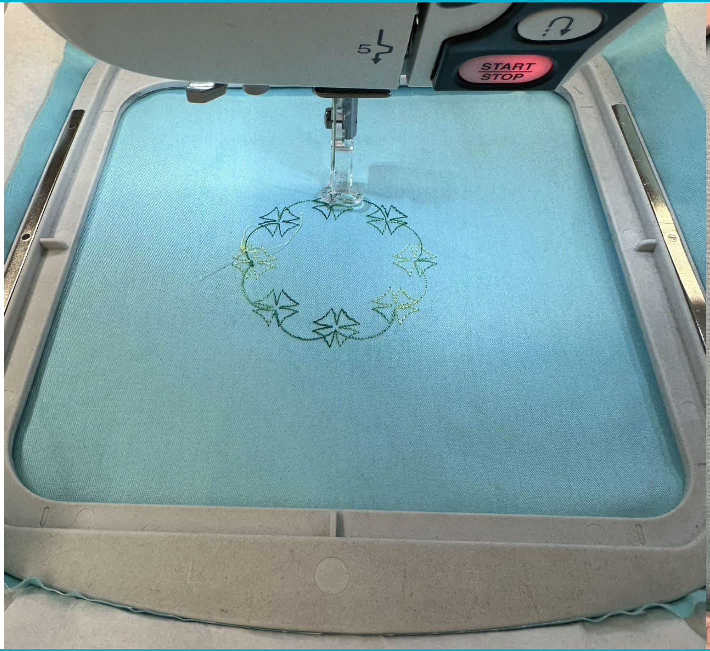
3/12/2023 14:00

Grace Hour: 201.5 + 10 = 211.5h

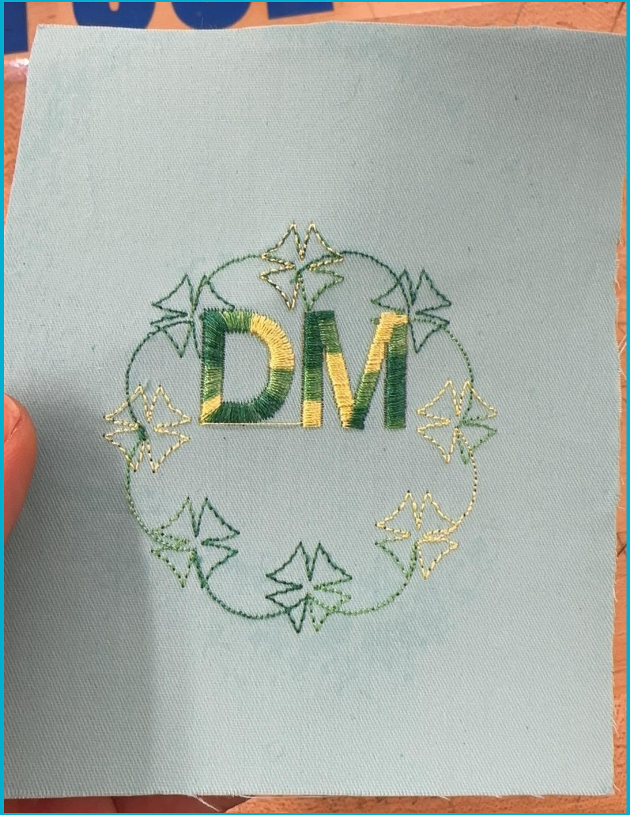


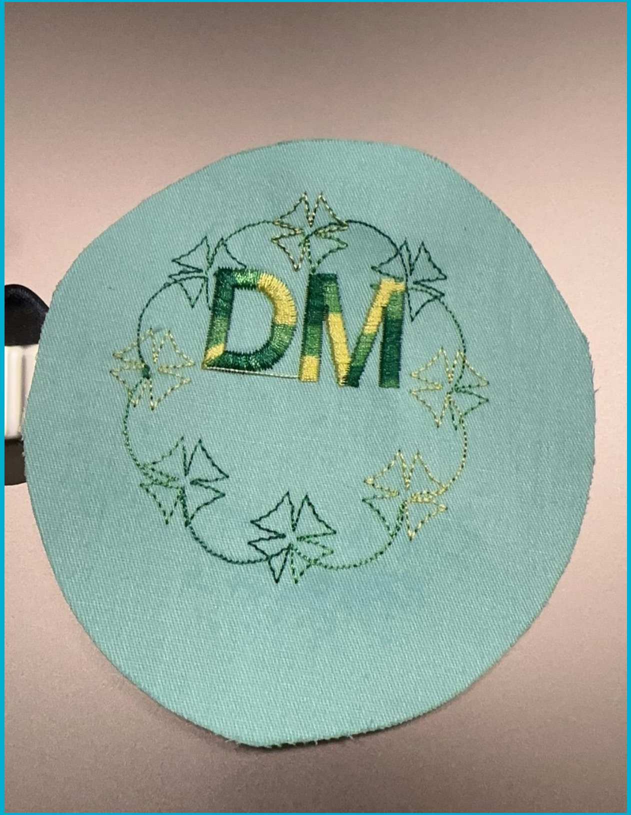
Embroidery Showcase





Embroidery on a Garment





Brief Procedure Description

The method used to create the embroidery design involved applying trigonometry to a circle with a size of radius 500. The process began by drawing a quarter circle using this circle-based unit and then repeating it to create a loop that formed a complete circle. By changing the direction of the quarter circle element every 90 degrees, the pattern was elongated to form an eight-patterned four-leaf clover shape. Each of the eight patterns created were then connected with a full circle. The use of trigonometry and a circle-based unit allowed for precision in the design process, resulting in a well-defined and structured final image. This method is commonly used in embroidery design as it allows for the creation of intricate designs with a high level of accuracy and detail. By utilizing programming languages like Python, this design process can be streamlined and automated, allowing for faster and more efficient creation of embroidery designs.

Code

```
import math

stitches = [0, 0]
stitches += [0, 0]
stitches += [206, 206]

theta0 = 0.0
N = 36
theta1 = math.pi/N
r = 500.0
dx_list = []
dy_list = []
d = 1000
stitches = [100, 100]

for i in range(0,72):
    dx = 0.5*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 0.5*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dx, dy]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)
for i in range(0,27):
    dx = 1*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 1*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dy, dx]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)

for i in range(0,72):
    dx = 0.5*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 0.5*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dx, dy]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)

for i in range(9,36):
    dx = 1*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
dy = 1*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dy, dx]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)

for i in range(0,72):
    dx = 0.5*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 0.5*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dx, dy]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)


for i in range(18,45):
    dx = 1*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 1*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dy, dx]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)

for i in range(0,72):
    dx = 0.5*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 0.5*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dx, dy]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)


for i in range(27,54):
    dx = 1*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 1*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
```

Code 2

```
if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dy, dx]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)

for i in range(0,72):
    dx = 0.5*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 0.5*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dx, dy]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)


for i in range(36,63):
    dx = 1*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 1*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
```

```
if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dy, dx]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)

for i in range(0,72):
    dx = 0.5*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 0.5*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dx, dy]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)


for i in range(45,72):
    dx = 1*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 1*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
```

```
if dy < 0:
        dy = 256+dy
    stitches += [dy, dx]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)

for i in range(0,72):
    dx = 0.5*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 0.5*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dx, dy]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)


for i in range(-18,9):
    dx = 1*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 1*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
```

```
stitches += [dy, dx]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)

for i in range(0,72):
    dx = 0.5*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 0.5*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dx, dy]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)

for i in range(-9,18):
    dx = 1*r*math.cos(theta1*
(i+1))-r*math.cos(theta1*i)
    dy = 1*r*math.sin(theta1*
(i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dy, dx]
    print(dx)
    print(dy)
    dx_list.append(dx)
```

Code 3

```
dy_list.append(dy)
    print(dx)
    print(dy)

for i in range(0,72):
    dx = 0.5*r*math.cos(theta1* (i+1))-r*math.cos(theta1*i)
    dy = 0.5*r*math.sin(theta1* (i+1))-r*math.sin(theta1*i)
    dx = round(dx)
    dy = round(dy)
    if dx < 0:
        dx = 256+dx
    if dy < 0:
        dy = 256+dy
    stitches += [dx, dy]
    print(dx)
    print(dy)
    dx_list.append(dx)
    dy_list.append(dy)
    print(dx)
    print(dy)


stitches += [128, 16]

jefBytes = [128, 0, 0, 0,  # The byte offset of the first stitch
             10, 0, 0, 0,  # unknown command
             ord("2"), ord("0"), ord("2"), ord("1"),  # YYYY
             ord("0"), ord("2"), ord("2"), ord("4"),  # MMDD
             ord("1"), ord("5"), ord("2"), ord("1"),  # HHMM
             ord("0"), ord("0"), 99, 0,  # SS00
             1, 0, 0, 0,  # Thread count nr. (nr of thread changes)
             (len(stitches)//2) & 0xff, (len(stitches)//2) >> 8 & 0xff, 0,
0,  # Number of stitches
             3, 0, 0, 0,  # Sewing machine Hoop
             # Extent 1
             50, 0, 0, 0,  # Left boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Top boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Right boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Bottom boundary dist from center (in 0.1mm)
```

```
# Extent 2
             50, 0, 0, 0,  # Left boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Top boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Right boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Bottom boundary dist from center (in 0.1mm)
             # Extent 3
             50, 0, 0, 0,  # Left boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Top boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Right boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Bottom boundary dist from center (in 0.1mm)
             # Extent 4
             50, 0, 0, 0,  # Left boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Top boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Right boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Bottom boundary dist from center (in 0.1mm)
             # Extent 5
             50, 0, 0, 0,  # Left boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Top boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Right boundary dist from center (in 0.1mm)
             50, 0, 0, 0,  # Bottom boundary dist from center (in 0.1mm)
             5, 0, 0, 0,  # Thread Color (white)
             2, 0, 0, 0,  # Thread Color (white)
             13, 0, 0, 0,  # Thread type (unknown)
             ] + stitches

jefBytes = bytes(jefBytes)
with open("DM STICH.jef", "wb") as f:
    f.write(jefBytes)
```

Points Achieved (140Pts)

10pts Cover page correct and complete

10pts Report neatly organized and formatted

10pt A parametric fractal shape embroidered

10pt Complexity/Aesthetics of the best pattern

10pt Quality of the stitch (over-stitching, wide stitches)

10pt Number of input parameters in software interface

10pt A description of the software you wrote – calculation steps, formulas, conditions.

10pt User specified text embroidered

10pt A fractal shape that is not a tree

10pt Multiple threads used (at least two)

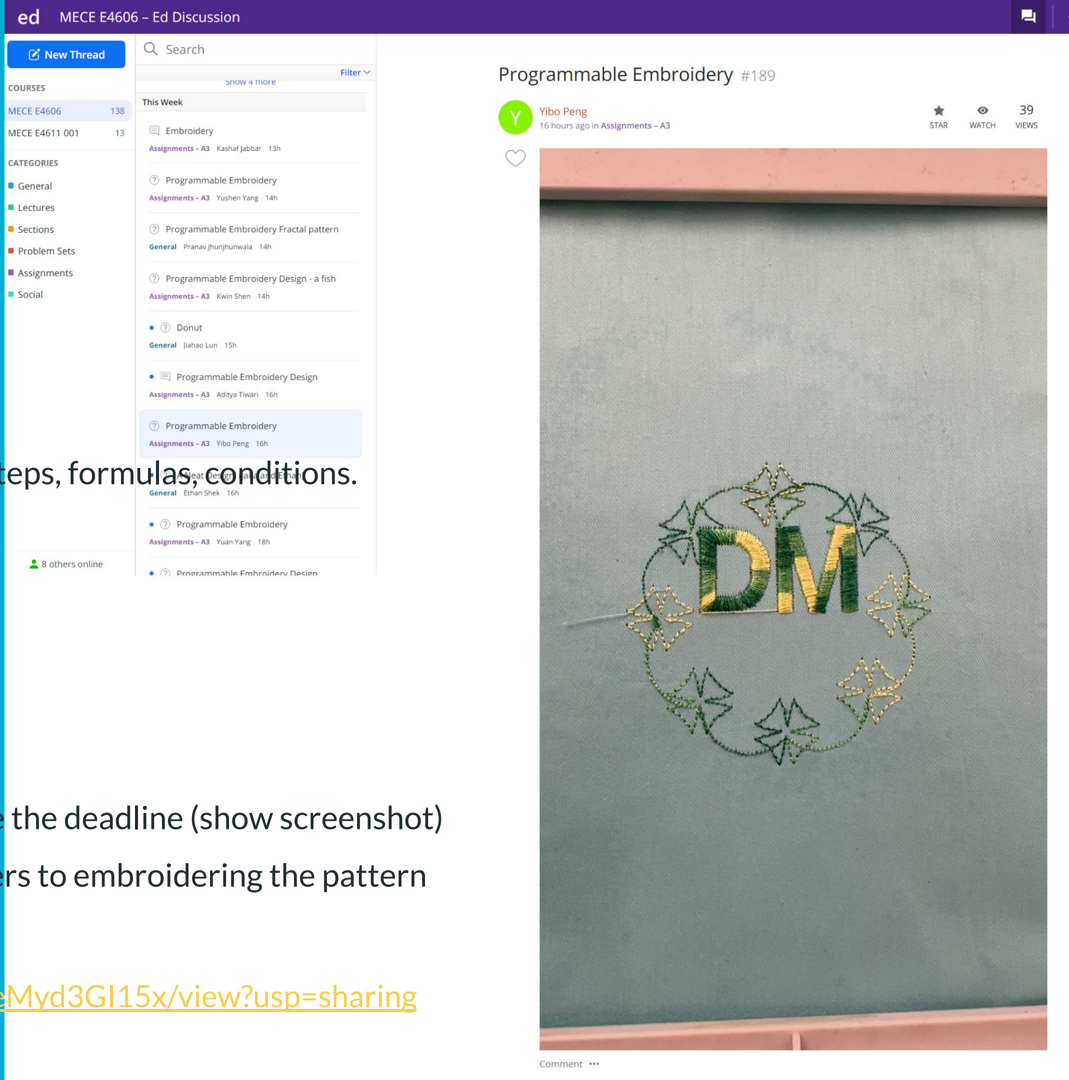
10pt Pattern sewn on a real garment

10pt Multiple threads colors used (excluding bobbin thread)

10pt Embroidery photo posted on Ed at least 24h day before the deadline (show screenshot)

10pt Video of entire process, from entering design parameters to embroidering the pattern

Video:
https://drive.google.com/file/d/1swpwuEYREigTg6rESZSVDFeMyd3GI15x/view?usp=sharing



MECE E4606 – Ed Discussion

COURSES

MECE E4606	138
MECE E4611 001	13

CATEGORIES

- General
- Lectures
- Sections
- Problem Sets
- Assignments
- Social

8 others online

This Week

Embroidery
Assignments – A3 Kashaf Jabbar 13h

Programmable Embroidery
Assignments – A3 Yushen Yang 14h

Programmable Embroidery Fractal pattern
General Pranav Jhunjhunwala 14h

Programmable Embroidery Design - a fish
Assignments – A3 Kwin Shen 14h

Donut
General Jiahao Lun 15h

Programmable Embroidery Design
Assignments – A3 Aditya Tiwari 16h

Programmable Embroidery
Assignments – A3 Yibo Peng 16h

A Neat Design [illegible] and Ethan
General Ethan Shek 16h

Programmable Embroidery
Assignments – A3 Yuan Yang 18h

Programmable Embroidery Design

Programmable Embroidery #189

Yibo Peng
16 hours ago in Assignments – A3

39 VIEWS