Programming in Python CST 362
Assignment 7

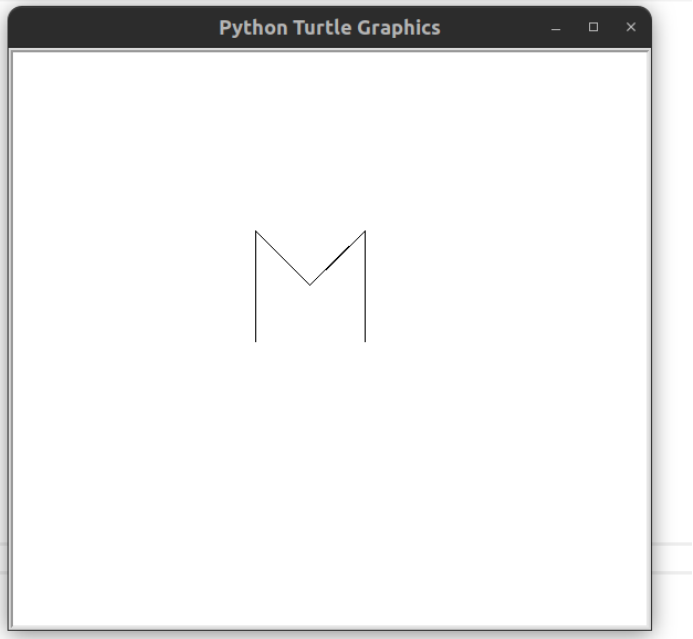
Mayon Francis
CS6A
44

1.Draw the first letter of your name using turtle graphics. (M)

1_M.py U

Assignment7 > 1_M.py > ...

```
1   from turtle import Turtle, done
2   t = Turtle()
3   t.penup()
4   t.goto(-70,0)
5
6   t.pendown()
7   t.left(90)
8   t.forward(100)
9   t.right(135)
10  t.forward(70)
11  t.left(90)
12  t.forward(70)
13  t.right(135)
14  t.forward(100)
15
16  t.up()
17  t.hideturtle()
18  done()
19
```

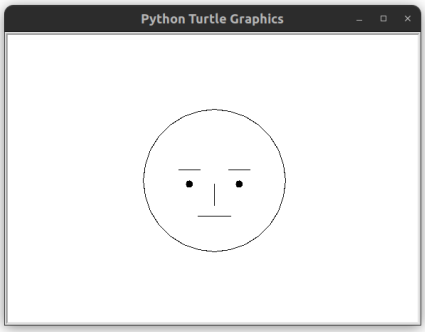


2. Draw the following figure (Face)

2_face.py U

Assignment7 > 2_face.py > ...

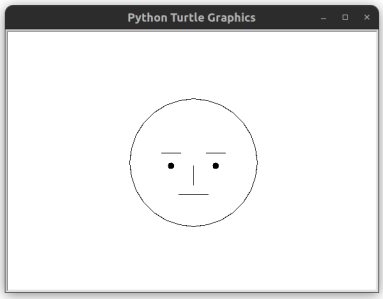
```
1   from turtle import Turtle, done
2
3   t = Turtle()
4   t.speed(0)
5
6   # Circle
7   start = (0,0)
8   lineLength = 30
9   radius = 100
10  t.up()
11  t.goto(start[0], start[1] - radius)
12  t.down()
13  t.circle(100)
14
15  # Nose
16  t.up()
17  t.goto(start[0], start[1] - lineLength//2 - 20)
18  t.down()
19  t.left(90)
20  t.forward(lineLength)
21
22  # Eyebrows
23  t.up()
24  t.forward(20)
25  t.left(90)
26  t.forward(lineLength + 20)
27  t.right(180)
28  t.down()
29  t.forward(lineLength)
30  t.up()
```



2_face.py U

Assignment7 > 2_face.py > ...

```
t.down()
t.forward(lineLength)
t.up()
t.forward(40)
t.down()
t.forward(lineLength)

# Left eye
t.up()
t.right(180)
t.forward(lineLength//2)
t.left(90)
t.forward(20)
t.down()
t.dot(10)

# Right eye
t.up()
t.right(90)
t.forward(lineLength + 40)
t.dot(10)

# Mouth
t.up()
t.goto(start[0]+lineLength*0.75, start[1] - radius //2)
t.down()
t.forward(lineLength*1.5)

t.hideturtle()
done()
```

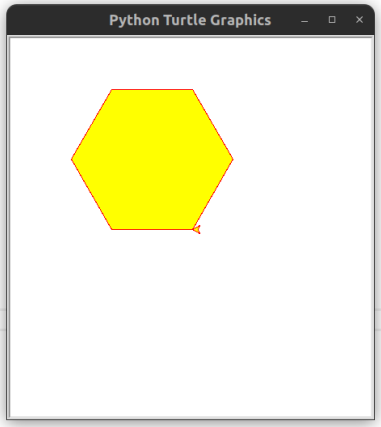


3.Draw a filled hexagon.(red color hexagon filled with yellow color)

3_hexagon.py U

Assignment7 > 3_hexagon.py > ...

```
from turtle import Turtle, done

t=Turtle()
t.left(180)

side = 100
angle = 60
t.color("red", "yellow")
t.begin_fill()
for i in range(6):
    t.forward(100)
    t.right(angle)

t.end_fill()
done()
```

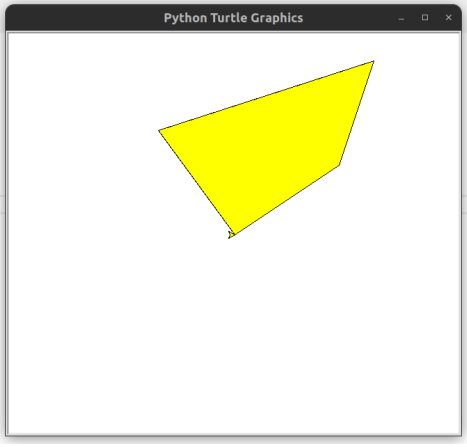


4.Draw a polygon. Coordinates of the polygon are stores in a list as tuples.

4_polygon.py U

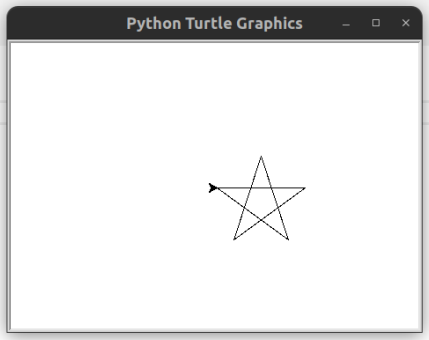
Assignment7 > 4_polygon.py > polygon

```
from turtle import Turtle, done
def polygon(t,poly):
    t.up()
    x,y=poly[-1]
    t.goto(x,y)
    t.down()
    t.fillcolor('yellow')
    t.pencolor('black')
    t.begin_fill()
    for x,y in poly:
        t.goto(x,y)
    t.end_fill()
t= Turtle()
poly=[(0,0),(150,100),(200,250),(-110,150),(0,0)]
polygon(t,poly)

done()
```



5. Draw a star

```python
import turtle
# s=turtle.getscreen()
t=turtle.Turtle()
for _ in range(5):
    t.forward(100)
    t.right(144)

turtle.done()
```

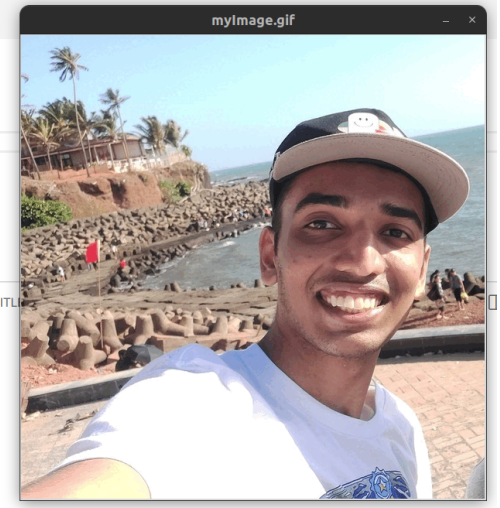


6.Take a selfy and create a gif file(use paint) and save it in your working directory as myimage.gif

7.Use image package shared earlier do the following Read and Display the image and image details

```python
from images import Image

img = Image('myImage.gif')
print(img)
img.draw()
```

```
Assignment7 (main): python3 7_selfie_1.py
File name: myImage.gif
Width:  600
Height: 600
```



Convert the image into black and white

```python
from images import Image

img = Image('myImage.gif')
black = (0,0,0)
white = (255,255,255)
threshold = 250

for y in range(img.getHeight()):
    for x in range(img.getWidth()):
        (r,g,b) = img.getPixel(x,y)
        avg = r+g+b // 3
        if avg < threshold:
            img.setPixel(x,y,black)
        else:
            img.setPixel(x,y,white)

img.draw()
```

Convert the image into gray scale

7_img_3_gray.py U

Assignment7 > 7_img_3_gray.py > ...

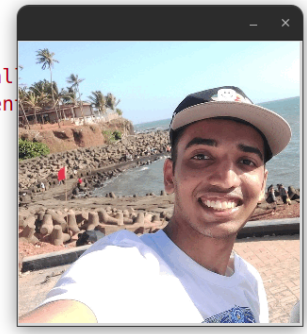
```
from images import Image

img = Image('myImage.gif')
black = (0,0,0)
white = (255,255,255)

for y in range(img.getHeight()):
    for x in range(img.getWidth()):
        (r,g,b) = img.getPixel(x,y)
        r = r * 0.299
        g = g * 0.587
        b = b * 0.114
        avg = int(r+g+b)
        img.setPixel(x,y,(avg,avg,avg))

img.draw()
```

Scale the image into one half

7_img_4_scale.py U

Assignment7 > 7_img_4_scale.py > ...

```
from images import Image
def shrink(image, factor):
    """Builds and returns a new image which is a smal
    copy of the argument image, by the factor argumen
    width = image.getWidth()
    height = image.getHeight()
    new = Image(width // factor, height // factor)
    oldY = 0
    newY = 0
    while oldY < height - factor:
        oldX = 0
        newX = 0
        while oldX < width - factor:
            oldP = image.getPixel(oldX, oldY)
            new.setPixel(newX, newY, oldP)
            oldX += factor
            newX += 1
        oldY += factor
        newY += 1
    return new

image=Image('myImage.gif')
print('Close the image to see the shrnked image')
image.draw()
image=shrink(image,2)
image.draw()
```

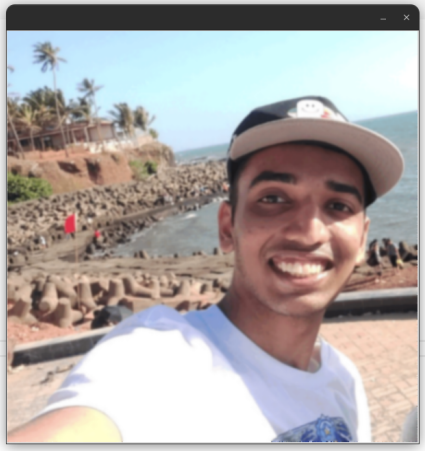


Blur the image

7_img_5_blur.py U

Assignment7 > 7_img_5_blur.py > blur

```
from images import Image
from  functools import *
def blur(image):
    """Builds and returns a new image which is a
    blurred copy of the argument image."""
    def tripleSum(triple1, triple2):
        (r1, g1, b1) = triple1
        (r2, g2, b2) = triple2
        return (r1 + r2, g1 + g2, b1 + b2)
    new = image.clone()
    for y in range(1, image.getHeight() - 1):
        for x in range(1, image.getWidth() - 1):
            oldP = image.getPixel(x, y)
            left = image.getPixel(x - 1, y) # To left
            right = image.getPixel(x + 1, y) # To right
            top = image.getPixel(x, y - 1) # Above
            bottom = image.getPixel(x, y + 1) # Below
            sums = reduce(tripleSum,[oldP, left, right, top, bottom])
            averages = tuple(map(lambda x: x // 5, sums))
            new.setPixel(x, y, averages)
    return new
image=Image('myImage.gif')
print('Close the image to see the blured image')
image.draw()
# Blurring multiple times to get a more blurred image
blurred = blur(image)
blurred = blur(blurred)
blurred = blur(blurred)
blurred = blur(blurred)
blurred = blur(blurred)
blurred.draw()
```



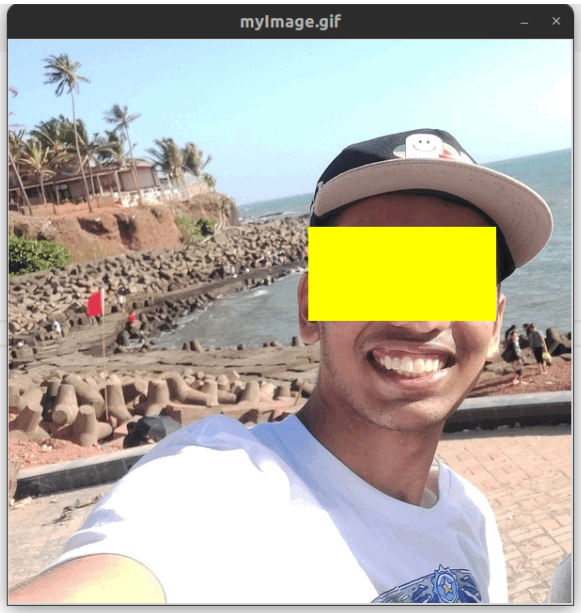
Place a block rectangle strip on the face

7_img_6_rect.py M

Assignment7 > 7_img_6_rect.py > ...

You, 26 seconds ago | 1 author (You)

```
from images import Image

img = Image('myImage.gif')

yellow = (255,255,0)
for x in range(320,520):
    for y in range(200,300):
        img.setPixel(x,y,yellow)
img.draw()      You, yesterday • Assignment 7 …
```

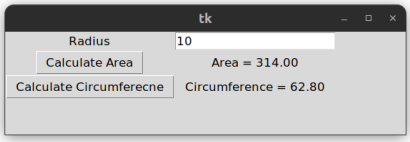


8. Create the GUI Applications using tkinter Read radius of a circle (use entry box to read) and find the area and circumference(use two separate command button and associated functions)

8_GUI_1_circle.py U

Assignment7 > 8_GUI_1_circle.py > ...

```python
from tkinter import Tk, StringVar, Label, Entry, Button, mainloop, W, DoubleVar
from datetime import date
def calcArea():
    areaVar.set(f"Area = {3.14*radiusVar.get()**2:.2f}")

def calcCircumference():
    circumVar.set(f"Circumference = {2*3.14*radiusVar.get():.2f}")

master=Tk()
radiusVar = DoubleVar()
areaVar = StringVar()
circumVar = StringVar()

Label(master, text="Radius").grid(row=0)
radiusEntry = Entry(master, textvariable=radiusVar).grid(row=0, column=1)

areaButton = Button(master, text="Calculate Area", command=calcArea)
areaButton.grid(row=1, column=0)
areaLabel = Label(master, textvariable=areaVar ).grid(row=1, column=1)

circumButton = Button(master, text="Calculate Circumferecne", command=calcCircumference)
circumButton.grid(row=2, column=0)
circumLabel = Label(master, textvariable=circumVar ).grid(row=2, column=1)

mainloop()
```



8. b) Write a GUI-based program that allows the user to convert temperature values between degrees Fahrenheit and degrees Celsius. The interface should have labeled entry fields for these two values. These components should be arranged in a grid where the labels occupy the first row and the corresponding fields occupy the second row. At start-up, the Fahrenheit field should contain 32.0, and the Celsius field should contain 0.0. The third row in the window contains two command buttons, labeled >>>> and <<<<. When the user presses the first button, the program should use the data in the Fahrenheit field to compute the Celsius value, which should then be output to the Celsius field. The second button should perform the inverse function.

8_GUI_2_TempConv.py U

Assignment7 > 8_GUI_2_TempConv.py > ...

```python
    celsiusVar.set(celsius)

def celsToFahr():
    fahren = (celsiusVar.get() * 9 / 5) + 32
    fahrenVar.set(fahren)

master=Tk()
fahrenVar = DoubleVar()
celsiusVar = DoubleVar()
fahrenVar.set(32.0)
celsiusVar.set(0)

Label(master, text="Fahrenheit").grid(row=0, column=1, pady=5)
Label(master, text="Celsius").grid(row=0, column=2, pady=5)
fahrenEntry = Entry(master, textvariable=fahrenVar).grid(row=1, column=1, padx=5)
celsiusEntry = Entry(master, textvariable=celsiusVar).grid(row=1, column=2, padx=5)

areaButton = Button(master, text=">>>>>>", command=fahrToCels)
areaButton.grid(row=2, column=1, pady=5)
areaButton = Button(master, text="<<<<<<", command=celsToFahr)
areaButton.grid(row=2, column=2, pady=5)

mainloop()
```