ORMEDIAN RESEARCH INSTITUTE

TOPIC: : OpenCV Tutorials (part four)

By:

ADEKOLA OLUWASEUN O.

February 21, 2023

TABLE OF CONTENT

- DRAWING & WRITING ON IMAGES IN OpenCV
- DRAWING OF BLANK IMAGES
- PAINTING OF INTERNAL PART OF BLANK IMAGES
- DRAWING AND PAINTING OF RECTANGLES
- DRAWING AND PAINTING OF CIRCLES
- DRAWING AND PAINTING OF TRIANGLES
- DRAWING AND PAINTING OF ELLIPSE
- DRAWING OF LINES
- WRITING OF TEXT ON BLANK IMAGES

DRAWING & WRITING ON IMAGES IN OpenCV

Drawing of Blank Image

In openCV, text can be written on any image, be it a blank image or an existing real image. With the help of numpy module, parameters can be set to achieve a drawing of a blank image.

```
import cv2 as cv
import numpy as np
blank = np.zeros((height, width), dtype ='uint8')
cv.imshow(“window title", blank)
cv.waitKey(0)
```

DRAWING & WRITING ON IMAGES IN OpenCV Cont’d

Algorithm Explanation

Line 1: Import the openCV module

Line 2: Import the numpy module

Line 3: Create a blank image with a numpy.zeros() function to return a new array of a given shape and type filled with zeros.

Line 4: Display the Image

Line 5: specify waitKey() which will be pressed to end the display.

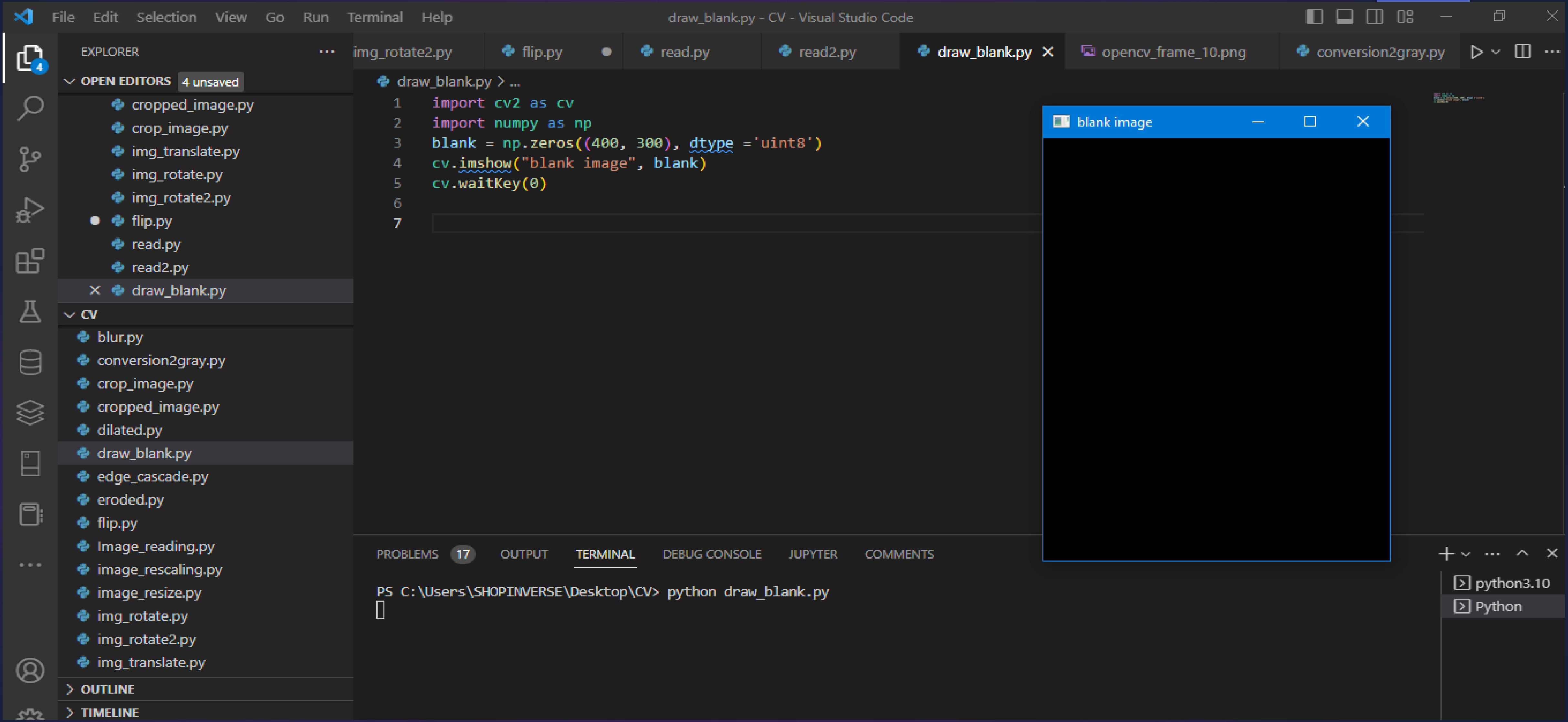
DRAWING & WRITING ON IMAGES IN OpenCV

Example: Draw a blank image of the following dimensions. Height = 400pixels and Width = 300 pixels.

SOLUTION

```
import cv2 as cv

import numpy as np

blank = np.zeros((400, 300), dtype ='uint8')

cv.imshow("blank image", blank)

cv.waitKey(0)
```

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((400, 300), dtype ='uint8')
cv.imshow("blank image", blank)
cv.waitKey(0)

```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python draw_blank.py
```

DRAWING & WRITING ON IMAGES IN OpenCV

Painting the Internal part of an Image in OpenCV

In OpenCV, blank images can be painted with different colours across all its areas or some parts of their areas. The only modification to the blank image created earlier is simply specifying the region to be painted.

SYNTAX

```
import cv2 as cv
import numpy as np
blank = np.zeros((height, width, colour channels), dtype ='uint8')
blank[specify regions to be paint] = specify colour
cv.imshow("blank image", blank)
cv.waitKey(0)
```

DRAWING & WRITING ON IMAGES IN OpenCV

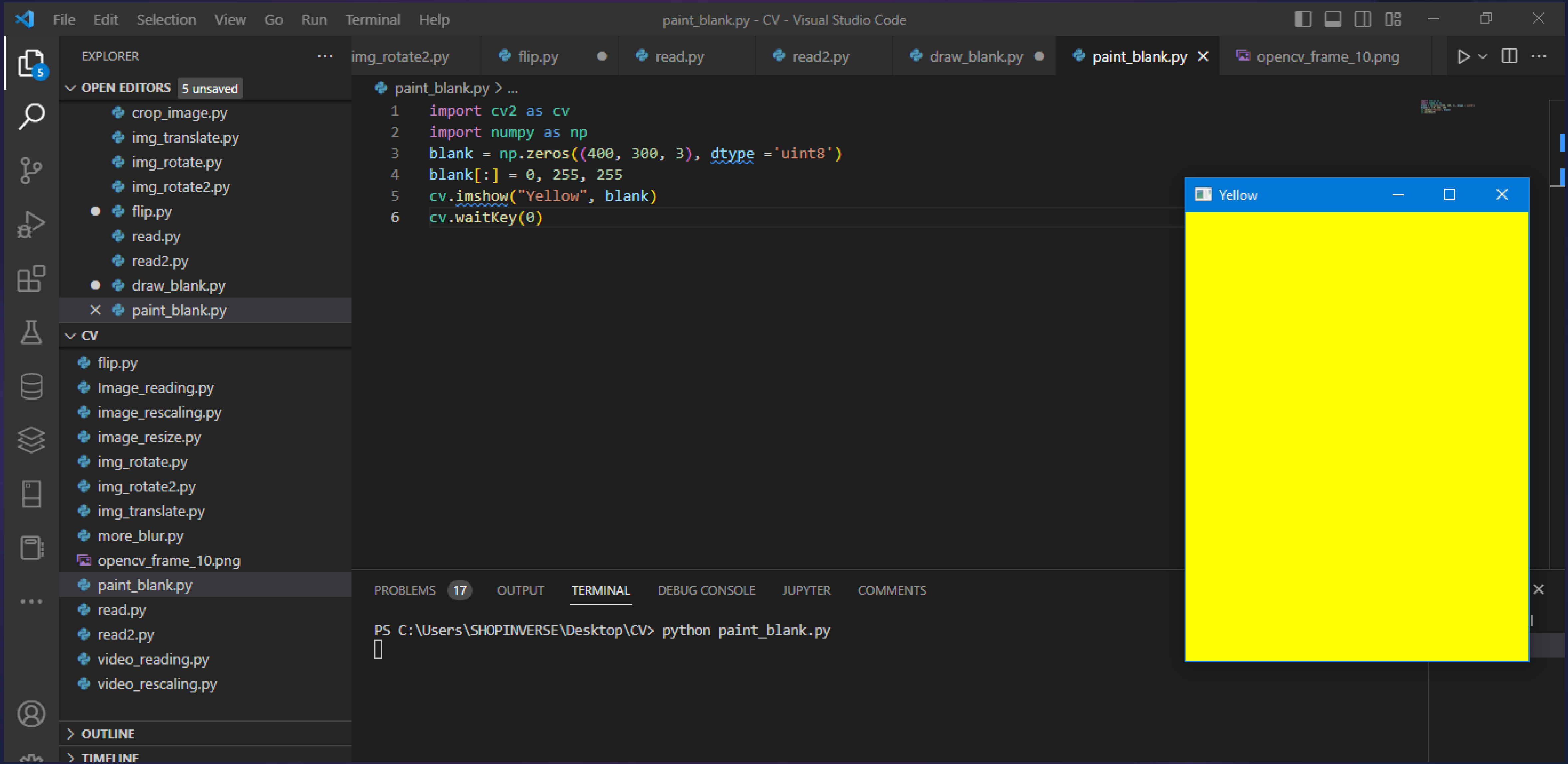
Example 1: Create a blank image of dimensions of choice and paint all its regions in any colour of choice.

SOLUTION

```
import cv2 as cv

import numpy as np

blank = np.zeros((400, 300, 3), dtype ='uint8')

blank[:] = 0, 225, 255

cv.imshow(“Yellow", blank)

cv.waitKey(0)
```

The blank [:] is simply the selection of all areas of the blank image created earlier, while the colour code (0, 225, 255) is a yellow-like colour which is an RGB colour category. And for such a reason, the 3 channels are passed into the shape formation.

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((400, 300, 3), dtype ='uint8')
blank[:] = 0, 255, 255
cv.imshow("Yellow", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python paint_blank.py
```

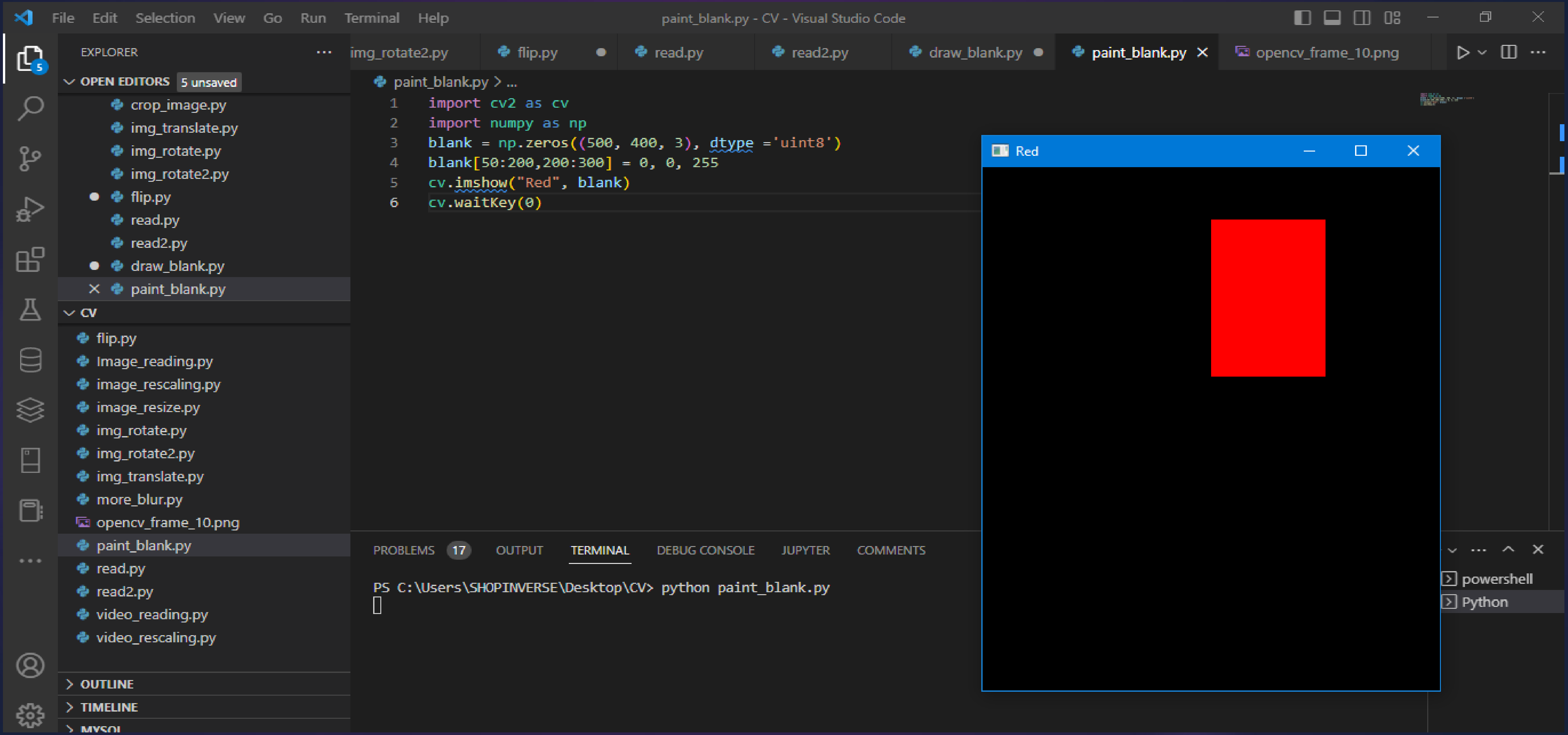
DRAWING & WRITING ON IMAGES IN OpenCV

Example 2: Create a blank image of dimensions of choice and paint some parts of its regions any colour of choice.

SOLUTION

```
import cv2 as cv
import numpy as np
blank = np.zeros((400, 300, 3), dtype ='uint8')
blank[50:200, 200:300] = 0, 0, 255
cv.imshow(“Red", blank)
cv.waitKey(0)
```

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
blank[50:200,200:300] = 0, 0, 255
cv.imshow("Red", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python paint_blank.py
```

DRAWING & WRITING ON IMAGES IN OpenCV

DRAWING OF A RECTANGLE IN OpenCV

Rectangle: Rectangle is a four-sided plain shape that requires some arguments to be specified for it in OpenCV.

SYNTAX:

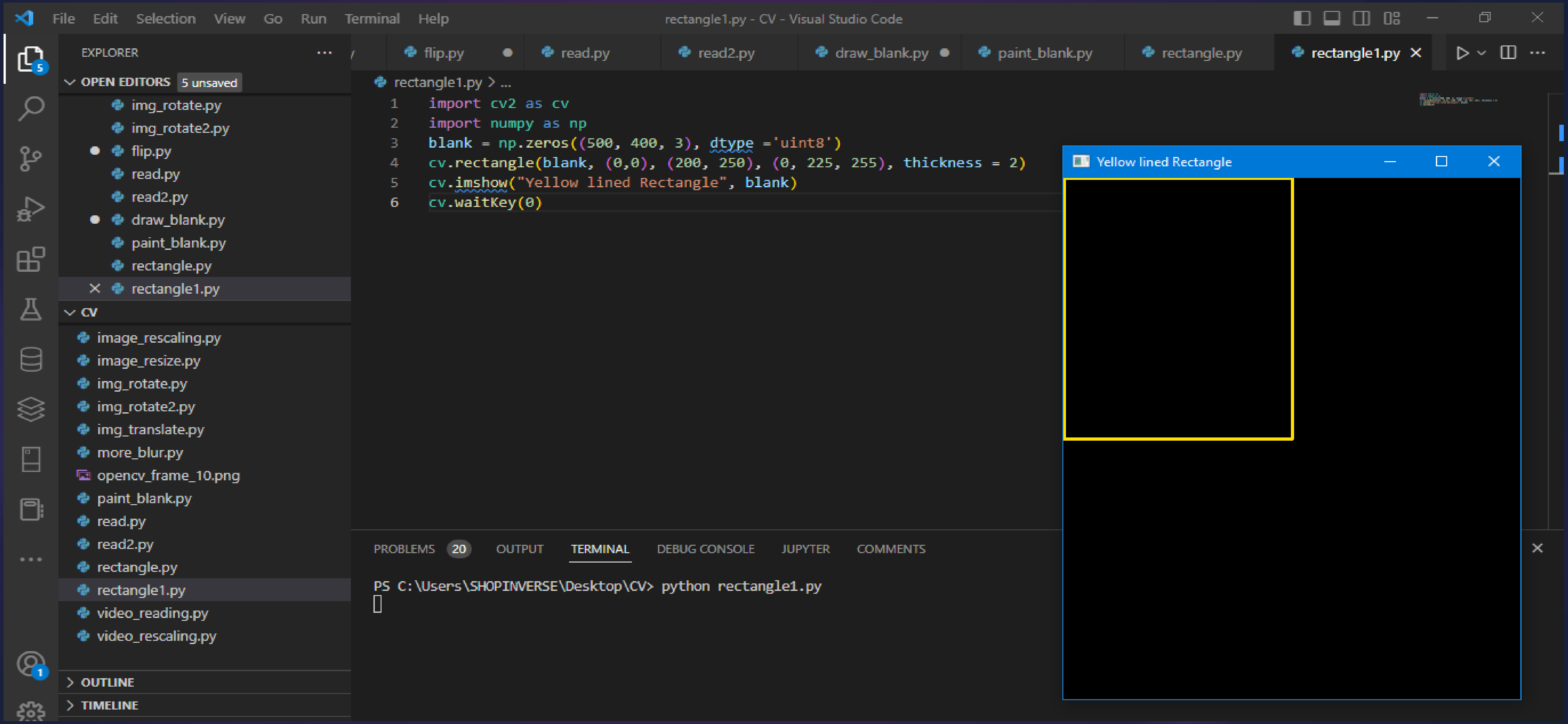
cv.rectangle (src, pt1, pt2, colour, thickness, linetype)

DRAWING & WRITING ON IMAGES IN OpenCV

Example 1: Draw a rectangular shape on a blank image such that the lines are of yellow colour. . Assume the dimension of the rectangle from the origin is 200 x 250 pixels, given that we still retain the blank images we have been using from previous sections.

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.rectangle(blank, (0,0), (200, 250), (0, 225, 255), thickness = 2)
cv.imshow("Yellow Rectangle", blank)
cv.waitKey(0)
```

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.rectangle(blank, (0,0), (200, 250), (0, 225, 255), thickness = 2)
cv.imshow("Yellow lined Rectangle", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python rectangle1.py
```

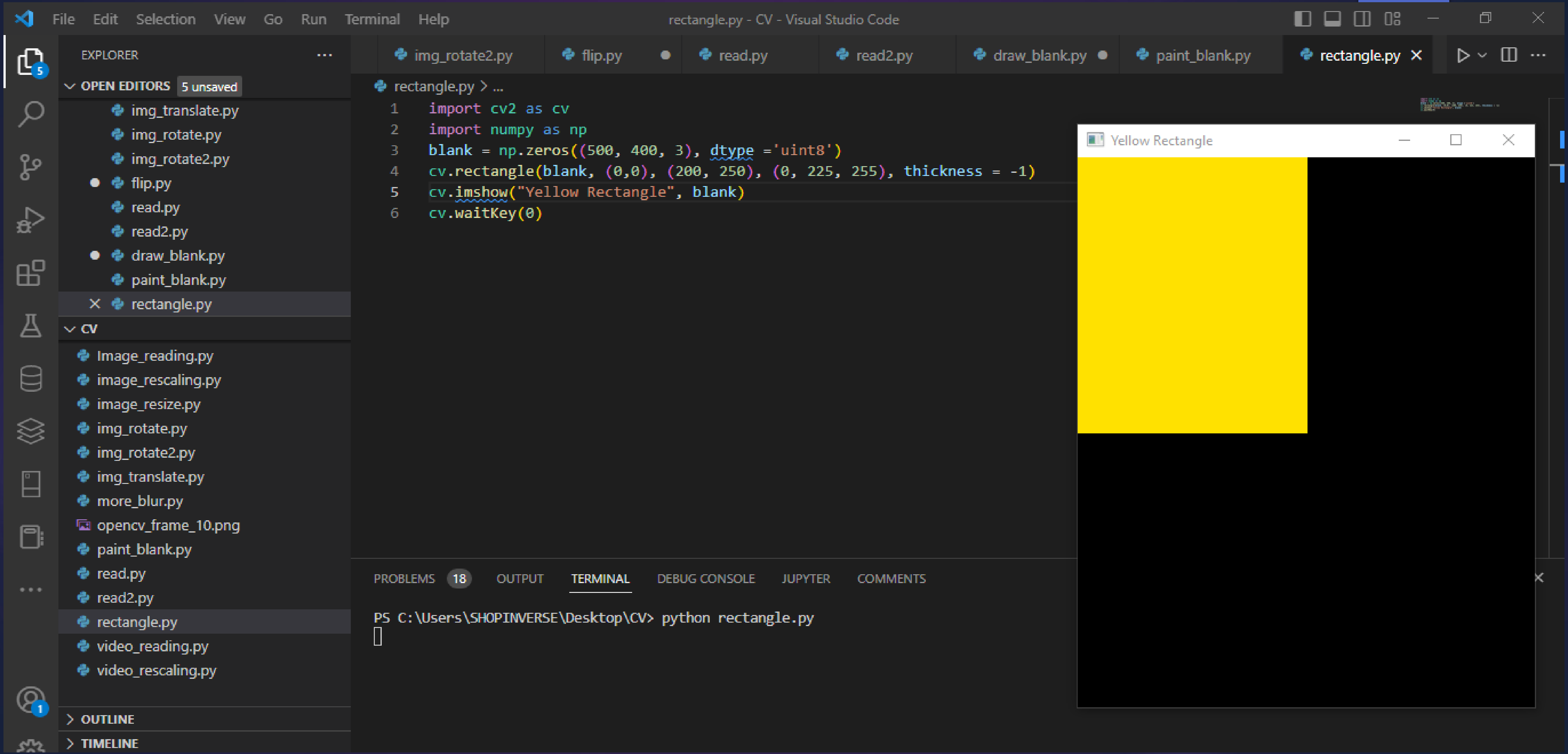
DRAWING & WRITING ON IMAGES IN OpenCV

Example 2: Draw a rectangular shape on a blank image and fill it all through with yellow colour. Assume the dimension of the rectangle from the origin is 250 x 200 pixels, given that we still retain the blank images we have been using from previous sections.

SOLUTION: The thickness parameter in this case can be set to = -1 or = cv.FILLED. This could alternatively be used to fill the shape with certain colour.

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.rectangle(blank, (0,0), (200, 250), (0, 225, 255), thickness = -1)
cv.imshow("Yellow Rectangle", blank)
cv.waitKey(0)
```

OUTPUT



```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.rectangle(blank, (0,0), (200, 250), (0, 225, 255), thickness = -1)
cv.imshow("Yellow Rectangle", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python rectangle.py
```

DRAWING & WRITING ON IMAGES IN OpenCV

DRAWING OF CIRCLE IN OpenCV

Circle: A circle is a plane figure whose circumference consists of points of equidistance from a fixed point called the center.

SYNTAX:

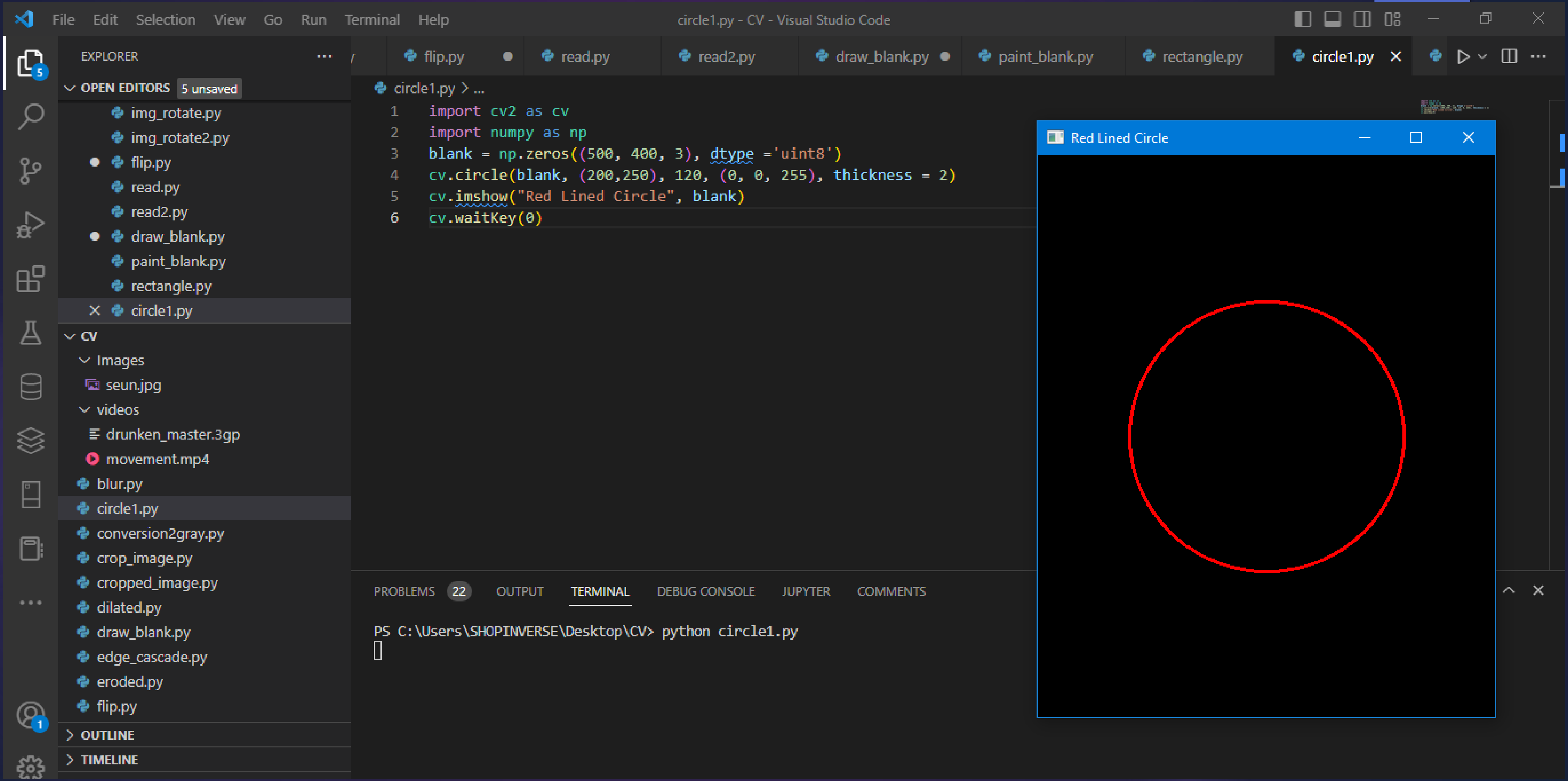
```
cv.circle (src, midpoint, radius, colour, thickness)
```

DRAWING & WRITING ON IMAGES IN OpenCV

Example 1: Draw a circle whose radius is 120 pixels, provided that the center of the circle is to be 200 x 250 pixels. Also, the line making the circumference of the circle should be red.

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.circle(blank, (200,250), 120, (0, 0, 255), thickness = 2)
cv.imshow("Red Lined Circle", blank)
cv.waitKey(0)
```

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.circle(blank, (200,250), 120, (0, 0, 255), thickness = 2)
cv.imshow("Red Lined Circle", blank)
cv.waitKey(0)
```

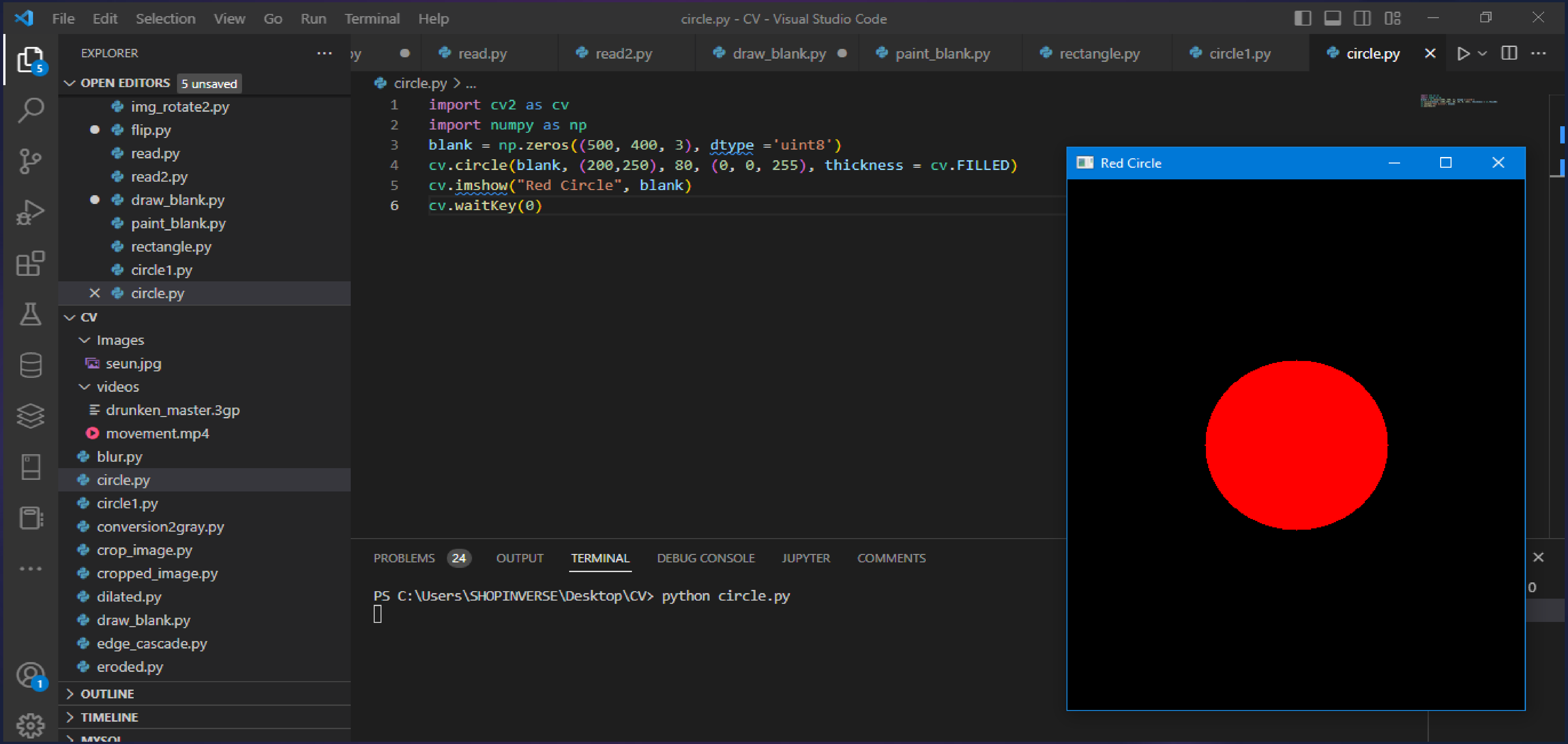
```
PS C:\Users\SHOPINVERSE\Desktop\CV> python circle1.py
```

DRAWING & WRITING ON IMAGES IN OpenCV

Example 2: Draw a circle whose radius is 80 pixels, provided that the center of the circle is to be 200 x 250 pixels. Also, the line making the circumference of the circle should be of red colour and also fill the shape with the same colour.

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.circle(blank, (200, 250), 80, (0, 0, 255), thickness = cv.FILLED)
cv.imshow("Red Circle", blank)
cv.waitKey(0)
```

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.circle(blank, (200,250), 80, (0, 0, 255), thickness = cv.FILLED)
cv.imshow("Red Circle", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python circle.py
```

DRAWING & WRITING ON IMAGES IN OpenCV

DRAWING OF A TRIANGLE IN OpenCV

Triangle: A triangle is a 3-sided figure. The cv.polylines functionality can be used to draw a triangle in OpenCV provided that the triangle vertices are specified.

Other parameters, such as src as the input image, pts as list of the array of points, isClosed to specify if the polyline should be closed or not, then colour and thickness should also be set.

To fill the internal part of such a shape with a particular colour, the cv.fillPoly() method is used to achieve that. The image, the bounded region, and the colour must be specified to implement this.

The use of the cv.polylines() method is demonstrated in example 1, while cv.fillPoly() method is demonstrated in example 2.

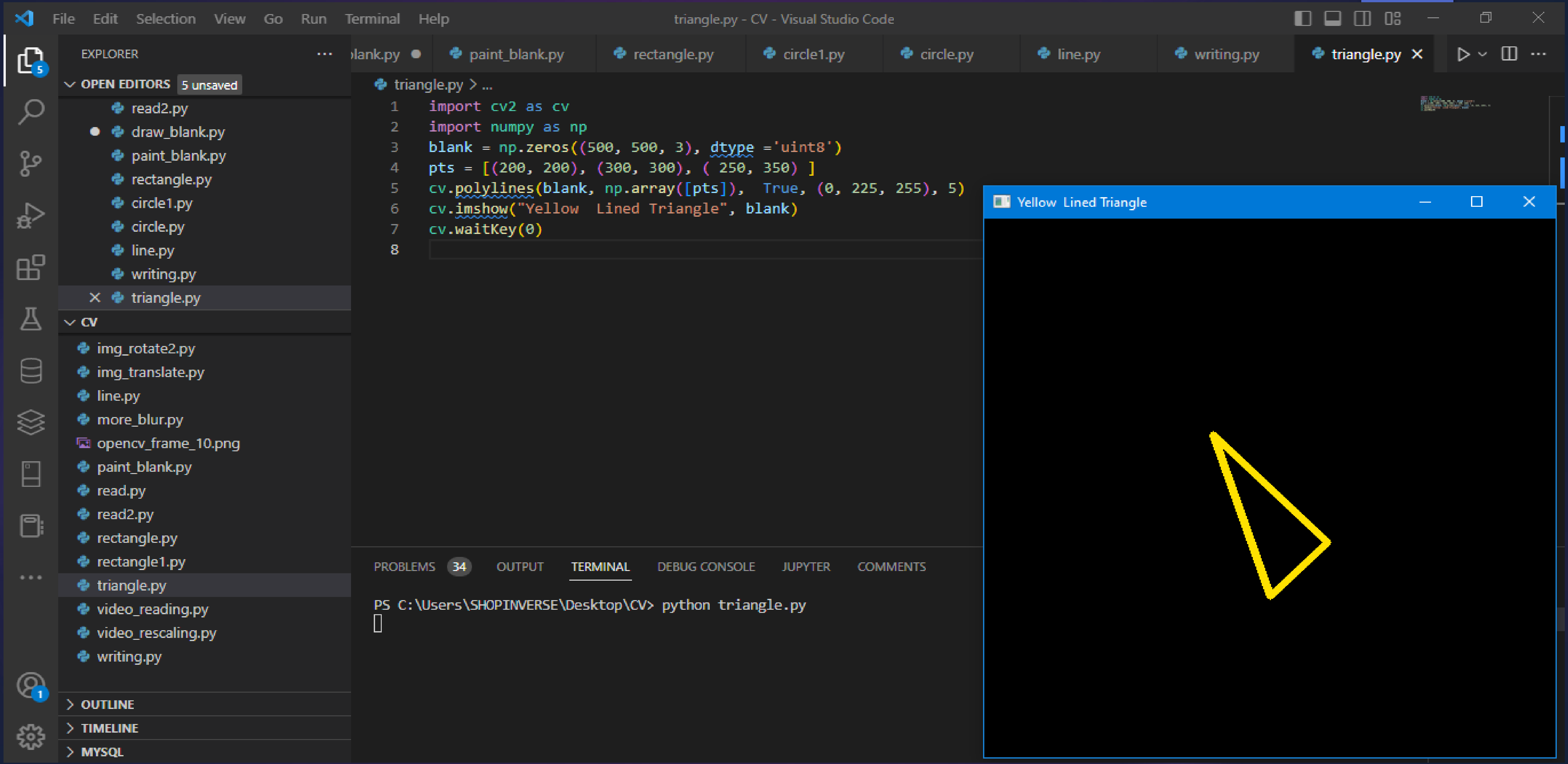
DRAWING & WRITING ON IMAGES IN OpenCV

Example 1: Draw a yellow lined triangle of the following with the following vertices (200, 200), (300, 300), (250, 350) on an existing blank image.

SOLUTION

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 500, 3), dtype ='uint8')
pts = [(200, 200), (300, 300), ( 250, 350) ]
cv.polylines(blank, np.array([pts]),  True, (0, 225, 255), 5)
cv.imshow("Yellow  Lined Triangle", blank)
cv.waitKey(0)
```

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((500, 500, 3), dtype ='uint8')
pts = [(200, 200), (300, 300), ( 250, 350) ]
cv.polylines(blank, np.array([pts]),  True, (0, 225, 255), 5)
cv.imshow("Yellow  Lined Triangle", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python triangle.py
```

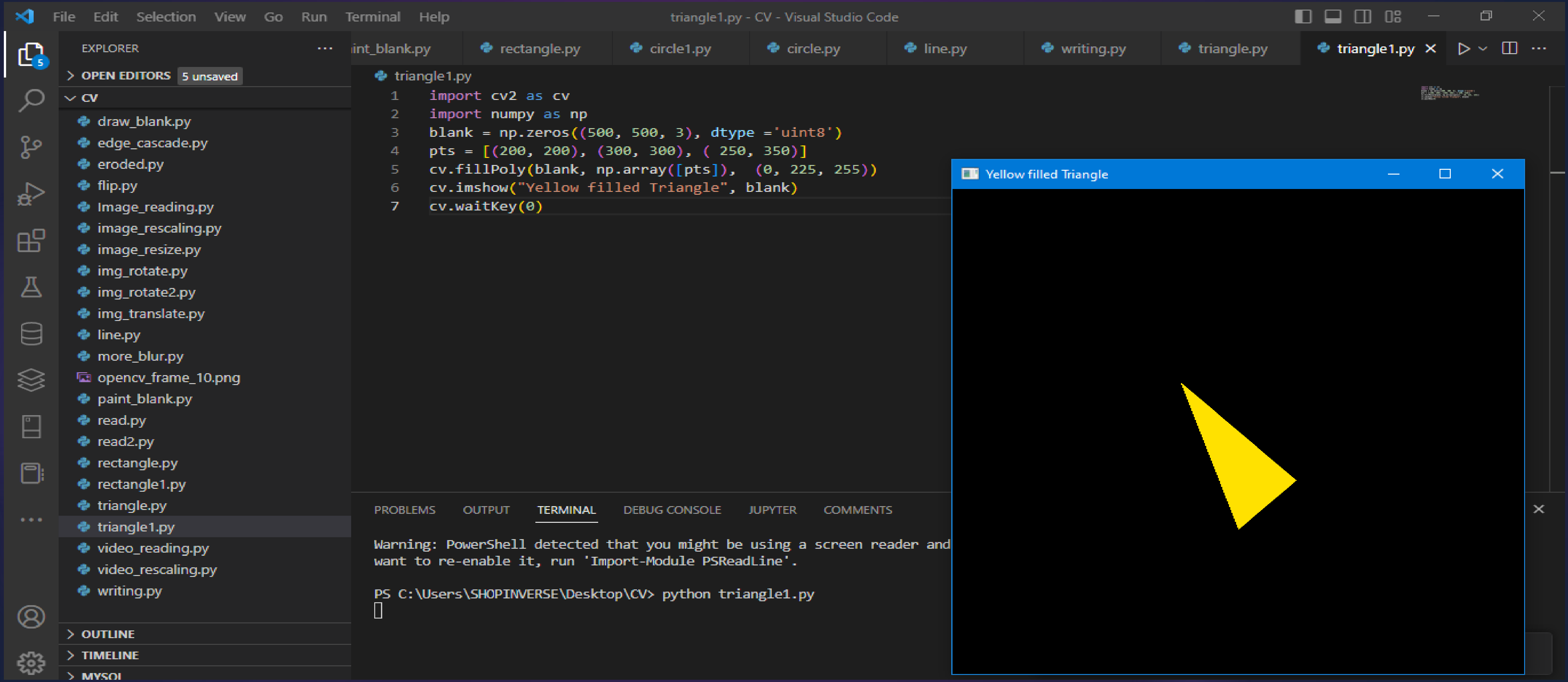
DRAWING & WRITING ON IMAGES IN OpenCV

Example 2: Draw a triangle with the following vertices (200, 200), (300, 300), (250, 350) on an existing blank image and fill it with a yellow colour.

SOLUTION

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 500, 3), dtype ='uint8')
pts = [(200, 200), (300, 300), ( 250, 350)]
cv.fillPoly(blank, np.array([pts]),  (0, 225, 255))
cv.imshow("Yellow filled Triangle", blank)
cv.waitKey(0)
```

OUTPUT



triangle1.py

```python
import cv2 as cv
import numpy as np
blank = np.zeros((500, 500, 3), dtype ='uint8')
pts = [(200, 200), (300, 300), ( 250, 350)]
cv.fillPoly(blank, np.array([pts]),  (0, 225, 255))
cv.imshow("Yellow filled Triangle", blank)
cv.waitKey(0)
```

```
Warning: PowerShell detected that you might be using a screen reader and
want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\SHOPINVERSE\Desktop\CV> python triangle1.py
```

DRAWING & WRITING ON IMAGES IN OpenCV

DRAWING OF AN ELLIPSE IN OpenCV

ELLIPSE: An llipse is a curve that is a locus of all points in the plane, the sum of whose distances R1 and R2 from two fixed points, F1 and F2 (the foci) separated by a distance. The OpenCV has inbuilt functionality for drawing an ellipse, which is shown in the syntax given below.

SYNTAX:

```
cv2.ellipse (src, CenterCoordinate, axesLength,
angle, startAngle, endAngle, colour, thickness,
lineType, shift)
```

DRAWING & WRITING ON IMAGES IN OpenCV

src: This is the input image on which the ellipse is to be drawn

centerCoordinates: This is the center of the ellipse

axesLength: This is a tuple used to specify the major axis and minor axis length.

Angle: This is the angle of rotation of the ellipse in degrees

startAngle: This is the starting angle of the elliptic arc in degrees

endAngle: The ending angle of the elliptic arc in degrees

Color: This is used to specify the border colour for the ellipse.

Thickness: This sets how thick the shape of the border should be.

Other parameters (lineType, shift etc.) are optional parameters.

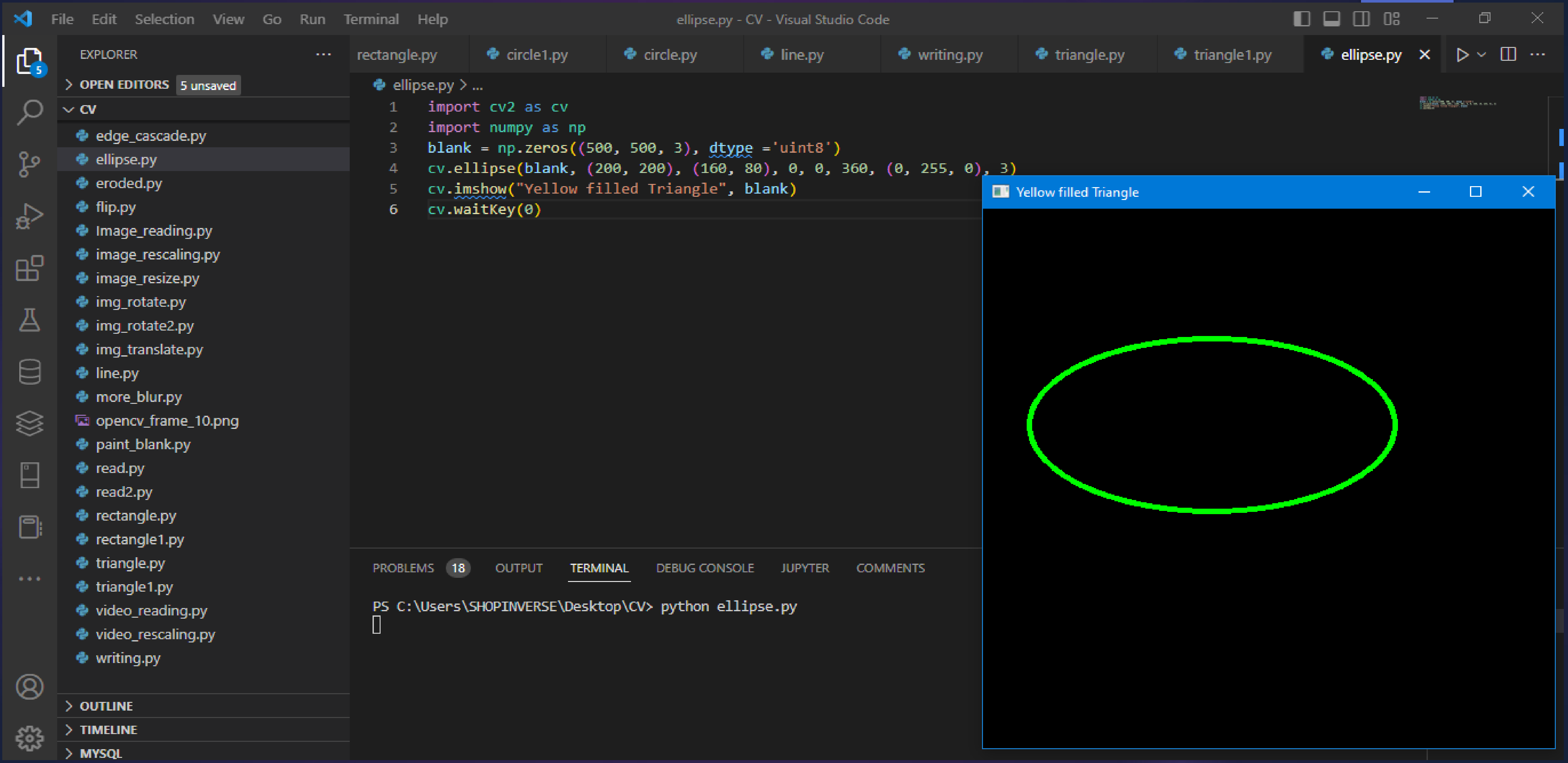
DRAWING & WRITING ON IMAGES IN OpenCV

Example 1: Draw an ellipse whose parameters are given as follows: major axis = 160, minor axis = 80, centerCoordinates = (200, 200), Angle = 0, startAngle = 0, endAngle = 360, colour = ‘GREEN’, thickness = 3.

SOLUTION

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 500, 3), dtype ='uint8')
cv.ellipse(blank, (200, 200), (160, 80), 0, 0, 360, (0, 255, 0), 3)
cv.imshow("Yellow filled Triangle", blank)
cv.waitKey(0)
```

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((500, 500, 3), dtype ='uint8')
cv.ellipse(blank, (200, 200), (160, 80), 0, 0, 360, (0, 255, 0), 3)
cv.imshow("Yellow filled Triangle", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python ellipse.py
```

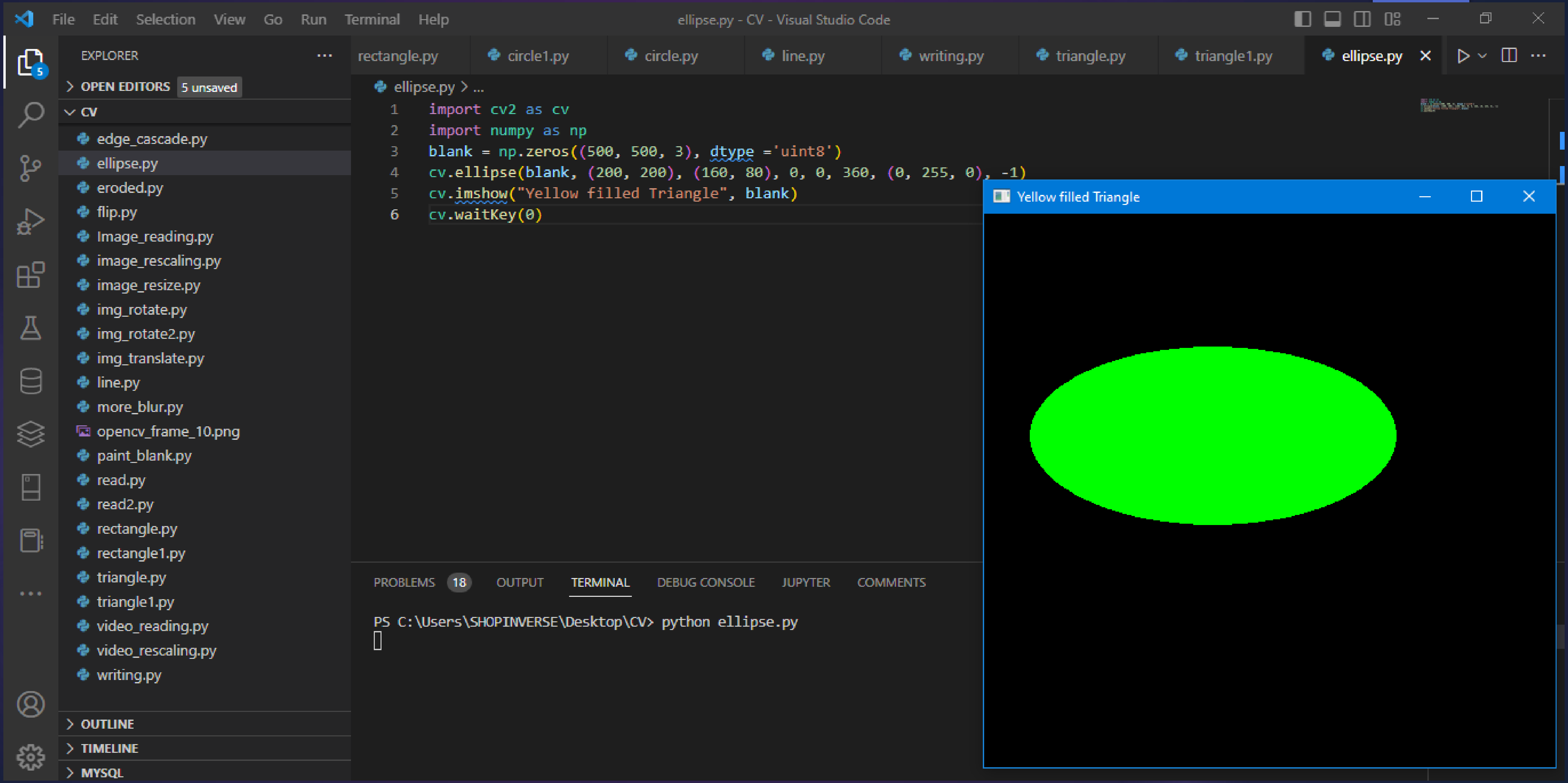
DRAWING & WRITING ON IMAGES IN OpenCV

Example 2: Draw an ellipse whose parameters are given as follows: major axis = 160, minor axis = 80, centerCoordinates = (200, 200), Angle = 0, startAngle = 0, endAngle = 360, colour = ‘GREEN’, thickness = -1.

SOLUTION

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 500, 3), dtype ='uint8')
cv.ellipse(blank, (200, 200), (160, 80), 0, 0, 360, (0, 255, 0), -1)
cv.imshow("Yellow filled Triangle", blank)
cv.waitKey(0)
```

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((500, 500, 3), dtype ='uint8')
cv.ellipse(blank, (200, 200), (160, 80), 0, 0, 360, (0, 255, 0), -1)
cv.imshow("Yellow filled Triangle", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python ellipse.py
```

DRAWING & WRITING ON IMAGES IN OpenCV

DRAWING OF A LINE IN OpenCV

Line: A line could be regarded as a straight one-dimensional figure that extends endlessly in both directions.

SYNTAX:

```
cv2.line (src, pt1, pt2, colour, thickness, linetype)
```

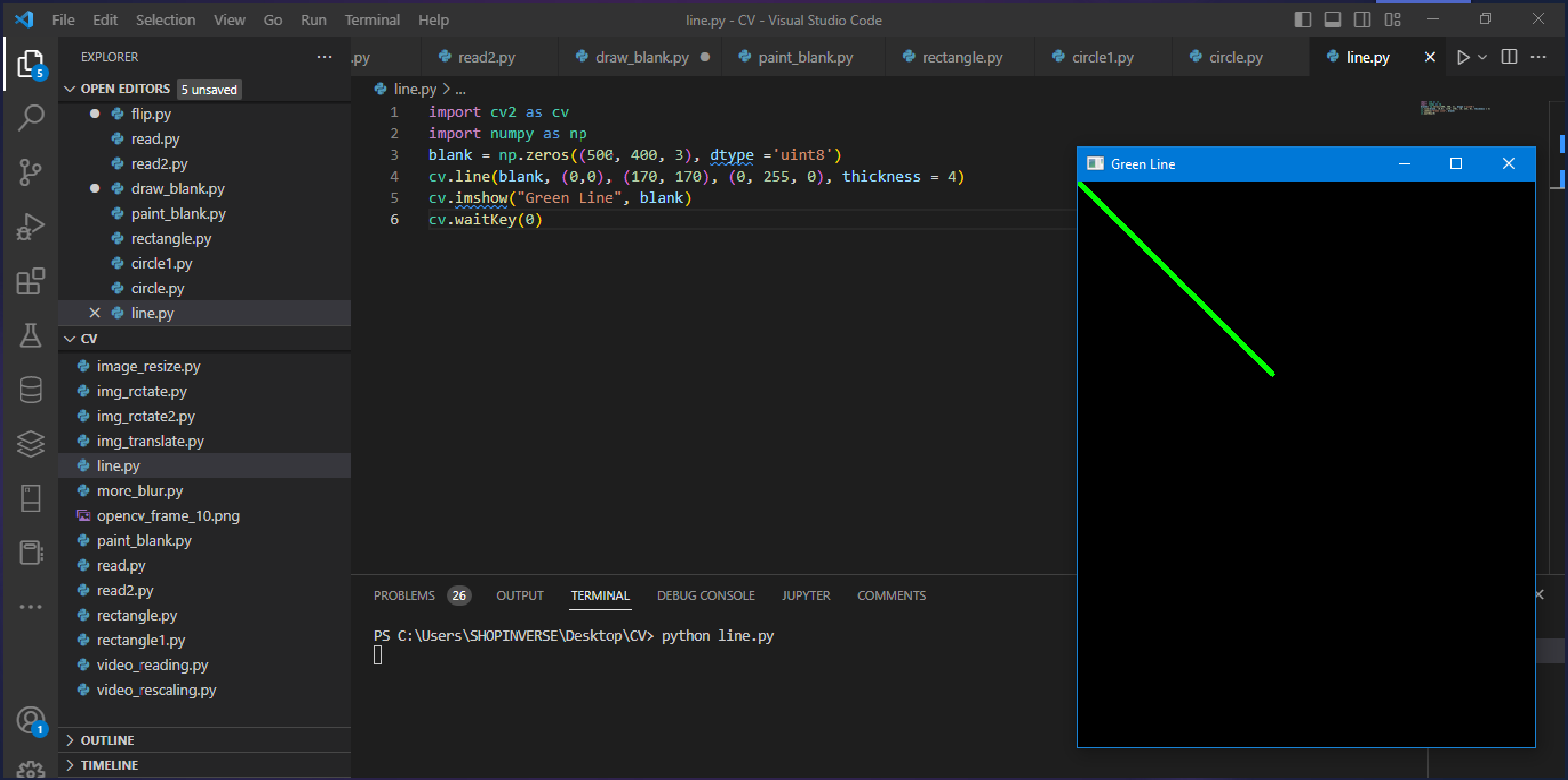
DRAWING & WRITING ON IMAGES IN OpenCV

Example 1: Draw a line whose origin is (0, 0) which extends to 170 pixels from the x- axis and 170 pixels from the y-axis. Provide such a line with a green colour of thickness of 4 pixels.

SOLUTION

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.line(blank, (0,0), (170, 170), (0, 255, 0), thickness = 4)
cv.imshow("Green Line", blank)
cv.waitKey(0)
```

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.line(blank, (0,0), (170, 170), (0, 255, 0), thickness = 4)
cv.imshow("Green Line", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python line.py
```

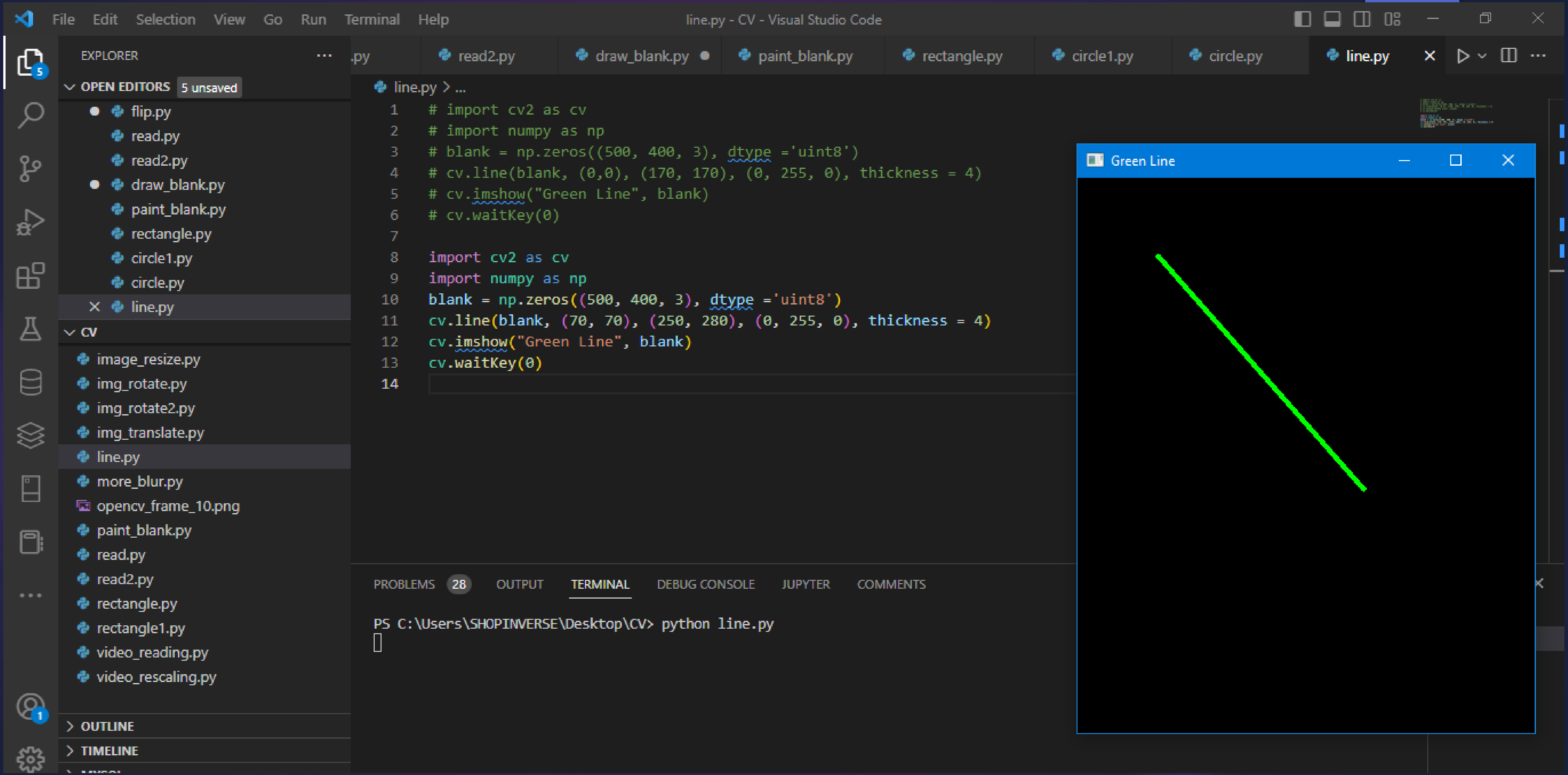
DRAWING & WRITING ON IMAGES IN OpenCV

Example 2: Draw a line whose origin is (70, 70) which extends to 250 pixels from the x- axis and 280 pixels from the y-axis. Provide such line with a green colour of thickness of 4 pixels.

SOLUTION

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.line(blank, (70, 70), (250, 280), (0, 255, 0), thickness = 4)
cv.imshow("Green Line", blank)
cv.waitKey(0)
```

OUTPUT



```python
# import cv2 as cv
# import numpy as np
# blank = np.zeros((500, 400, 3), dtype ='uint8')
# cv.line(blank, (0,0), (170, 170), (0, 255, 0), thickness = 4)
# cv.imshow("Green Line", blank)
# cv.waitKey(0)

import cv2 as cv
import numpy as np
blank = np.zeros((500, 400, 3), dtype ='uint8')
cv.line(blank, (70, 70), (250, 280), (0, 255, 0), thickness = 4)
cv.imshow("Green Line", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python line.py
```

DRAWING & WRITING ON IMAGES IN OpenCV

WRITING OF TEXT ON AN IMAGE

In OpenCV text can be written on a blank or normal image. To achieve this, some parameters have to be set. This involves specifying the image to be written on, the text to write on it, the area of the blank image to start writing from, font style, scaling, colour, and thickness. The cv.putText() method is used to implement this.

In the process of specifying arguments for these parameters, it is important to note that OpenCV has some inbuilt font style which can be selected from.

SYNTAX

```
cv2.putText(blank,  text, text origin, font style, scaling, colour, thickness)
```

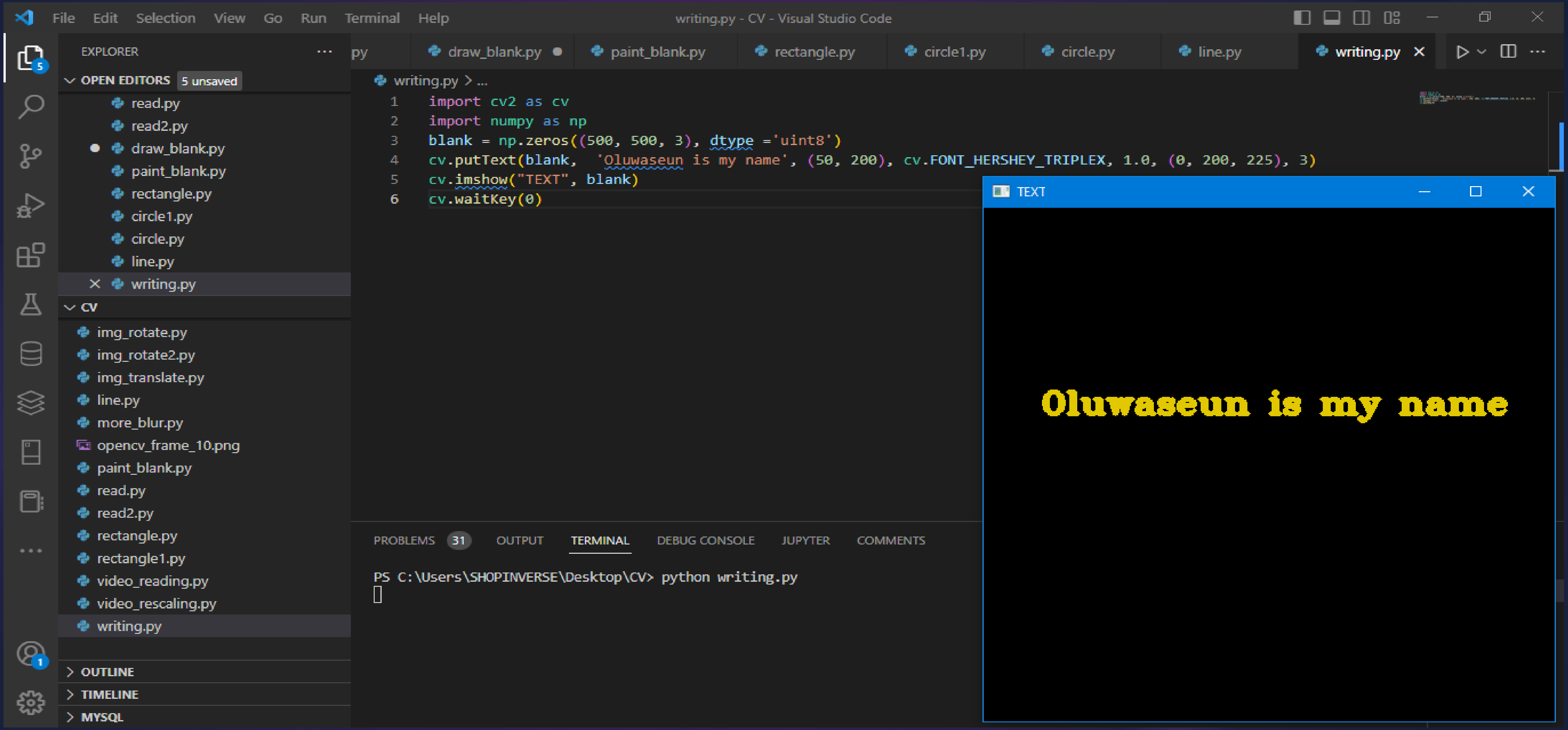
DRAWING & WRITING ON IMAGES IN OpenCV

EXAMPLE: Write a simple text on a blank image by following a well-defined rule of putting text on an image.

SOLUTION

```
import cv2 as cv
import numpy as np
blank = np.zeros((500, 500, 3), dtype ='uint8')
cv.putText(blank,  'Oluwaseun is my name', (50, 200),
cv.FONT_HERSHEY_TRIPLEX, 1.0, (0, 200, 225), 3)
cv.imshow("TEXT", blank)
cv.waitKey(0)
```

OUTPUT



```python
import cv2 as cv
import numpy as np
blank = np.zeros((500, 500, 3), dtype ='uint8')
cv.putText(blank,  'Oluwaseun is my name', (50, 200), cv.FONT_HERSHEY_TRIPLEX, 1.0, (0, 200, 225), 3)
cv.imshow("TEXT", blank)
cv.waitKey(0)
```

```
PS C:\Users\SHOPINVERSE\Desktop\CV> python writing.py
```

Oluwaseun is my name

THANKS FOR VIEWING

More tutorials will be covered in part five