

## DAILY ASSESSMENT FORMAT

Date:	30/05/2020	Name:	Akshatha M Deshpande
Course:	Live Youtube Session	USN:	4AL17EC006
Topic:	Simplifying the Brain	Semester & Section:	6th Sem A sec
Github Repository:	AkshathaDeshpande		

SESSION DETAILS	
Image of session	
	<p><b>NPTEL</b> Date: May 29, 2020 Time: 6.00 PM <a href="https://nptel.ac.in/">https://nptel.ac.in/</a></p>

Date:	30/05/2020	Name:	Akshatha M Deshpande
Course:	Logic Design	USN:	4AL17EC006
Topic:	Applications of programmable logic controllers	Semester & Section:	6th Sem A sec
Github Repository:	AkshathaDeshpande		

#### FORENOON SESSION DETAILS

Image of session

# An Introduction to Programmable Logic Controllers



programs look a lot like this now we're  
going to make this a little more

Report – Report can be typed or hand written for up to two pages.

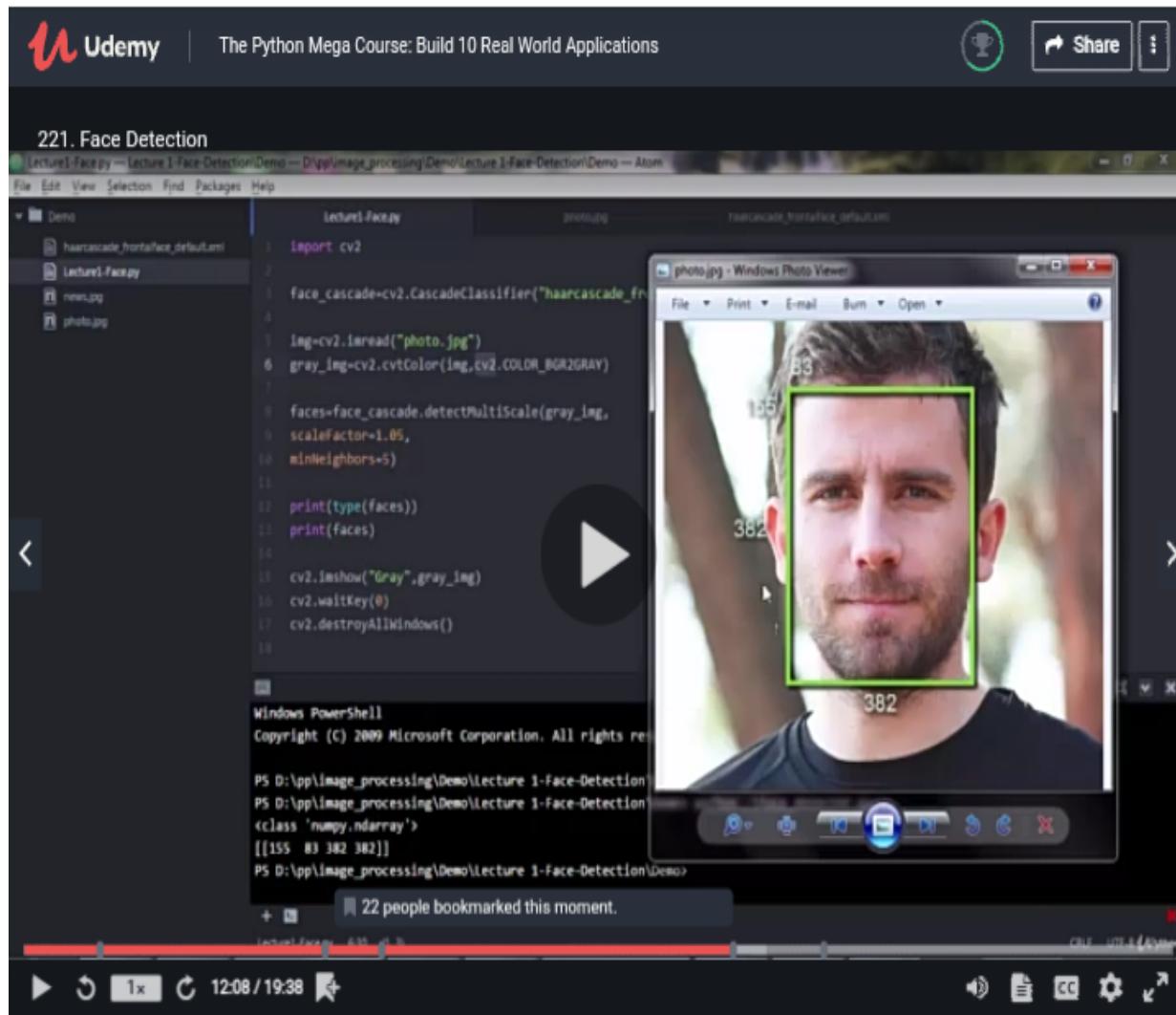
### Applications of programmable logic controllers:-

- \* Programmable logic controllers are the major component in industrial control systems they replaced something called Electro Magnetic Relays.
- \* The language used to program programmable controllers still uses symbology from the evolution from these relays to using computers to control manufacturing systems so this section we are fitting relays two bits basically making the transition from using relays to using bits in memory.
- \* A brief of history is needed to build a background for some of the terminology and symbols used in programming programmable logic controllers.
- \* Relay Diagram into an actual appreciation an actual circuit we normally refer to that portion of the circuit which interacts with the operator as the public interface.
- \* There are three public interfaces to this circuit the single pole single throw switch which the operator controls & when the closes the switch that energizes the relay when it opens it relieved the relay when it is energized then you have a red indicator & the battery powers the red LED when the relay is de-energized & then of course we have a green Indicator which will illuminate when the relay is energized.

Date: 30/05/2020 Name: Akshatha M Deshpande  
Course: Python USN: 4AL17EC006  
Topic: Python for Image and Semester & 6th Sem A sec  
Video Processing using OpenCv Section:

### AFTERNOON SESSION DETAILS

#### Image of session



**Report – Report can be typed or handwritten for up to two pages.**

## Python for image & video processing with opencv :-

- \* A image or a video can be read using open CV.
- \* How to install cv2

### Loading, Displaying, Resizing & writing images :-

- \* print(img) # Give the dimensions of entire matrix
- \* imread is used to read image after installing cv.
- \* print(type(img)) # gives type of image
- \* print(img.shape) # gives no. of rows & columns
- \* print(img.ndim) # gives dimension of image.
- \* zero gives grey image
- \* One gives colour image with 3 dimensions.
- \* To display image → imshow("Galaxy path", img) is used.
- \* Time can also be specified cv.waitKey(0).

Any button pressed image will close.

cv2.waitKey(2000) # waits

cv2.destroyAllWindows() # deletes

- \* The image used here is 1488 pixels height, 998 pixels width.

To resize, re-img = cv2.resize(img, (1000, 500)) <sup>Galaxy path</sup>. take this.

We then store the resized image using imwrite.

Many images can be serialized using \*.jpg.

### Face detection :-

- \* The pixels are stored in XML files.
- \* OpenCV is used to create XML files.
- \* Grey scale provides more accuracy in detecting faces.



- \* gray-img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)
  - This converts color image to gray image.
- \* cv2.imshow("Gray", gray-img)
  - This shows the image
  - Wait and then destroy the windows.

⇒ Detect Multiscale :-

- \* This searches for XML file & cascade classifier.
- \* This returns the row & column of the face.
- \* faces = face\_cascade.detectMultiScale(gray-img,  
 scaleFactor=1.05,  
 minNeighbors=5)  
 print(faces)
- \* The type of face is detected by using  
 type(face).
- \* This gives the dimensions of face.
- \* For loop is used,  
 for x,y,w,h in faces:  
 img = cv2.rectangle(img, (x,y), (x+w),  
 y+h), (0,255,0), 3)  
 width  
 of rectangle.
- \* cv2.imshow(img)  
 # gives the image with a green color rectangle  
 on the mentioned dimension.



Edit with WPS Office