Practical 1:

Making a Raspberry Pi headless, and reaching it from the network using WiFi and SSH

https://youtu.be/HTl8NELUwUM?si=ik0PNPMH7M_nO9XP

Practical 2:

Using sftp upload files from PC. (Requires steps of practical 1 to be done first)

https://youtu.be/oUJxuKNrprM?si=r7PidGCs_KMbIED7

Practical 3:

Write Python code to test motors.

https://youtu.be/yliZvw5ffd8?si=pVn7ehpXwgShwvOv

Practical 4:

Develop Python code for testing the sensors.

https://prasadpusalkar.blogspot.com/2023/05/develop-script-to-test-sensors-ir-sensor.html?m=1

Practical 5:

Add the sensors to the Robot object and develop the line-following behaviour code.

https://youtu.be/eZY5aGMd7_M?si=RiyvZbTEAe6zF1oj

Practical 6:

Add pan and tilt service to the robot object and test it.

https://prasadpusalkar.blogspot.com/2023/05/add-pan-and-tilt-service-to-robot-object.html?m=1

Practical 7:

Create obstacle avoidance behaviour for robot and test it.

Part I **-** https://www.youtube.com/watch?v=7UmsW1c6K-A

Part II **-** https://youtu.be/lAcpBA8JqTA?si=KOqxU5b_ho25YcCA

Practical 8:

Detect faces with Haar cascades. (Python IDLE/Google Collab)

```
import cv2

face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')

image_path = 'faces.jpeg'
image = cv2.imread(image_path)

if image is None:
    print("Error: Could not load the image.")
    exit()

gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))

for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.imshow("Face Detection", image)

cv2.waitKey(0)
cv2.destroyAllWindows()
```