```
#!/usr/bin/env python


#A Turtlebot script that displays the image feeds

import rospy
from geometry_msgs.msg import Twist
from sensor_msgs.msg import Image
import cv2
from cv_bridge import CvBridge


class DisplayData():
        def __init__(self):
                # initiliaze
                rospy.init_node('DisplayData', anonymous=False)
                cv2.namedWindow("Image Window")
                cv2.namedWindow("Depth Window")
                # What function to call when you ctrl + c
                rospy.on_shutdown(self.shutdown)
                self.bridge = CvBridge()
                rospy.Subscriber("/camera/rgb/image_color/", Image,self.rbgCallback)
                rospy.Subscriber("/camera/depth/image/", Image, self.depthCallback)
                #rospy.spin() tells the program to not exit until you press ctrl + c.  If this wasn't
there... it'd subscribe to /laptop_charge/ then immediatly exit (therefore stop "listening" to the
thread).
                rospy.spin();


        def rbgCallback(self, data):
                rospy.loginfo("flagCallbackRGB")
                #Convert to openCV image type
                cv_image = self.bridge.imgmsg_to_cv2(data, 'bgr8')
                cv2.imshow("Image Window", cv_image)
                cv2.waitKey(1)

        def depthCallback(self, data):
                rospy.loginfo('flagCallbackDepth')
                cv_image = self.bridge.imgmsg_to_cv2(data)
                dst = cv_image
                cv_image = cv2.normalize(cv_image, dst, 0, 1, cv2.NORM_MINMAX)
                cv2.imshow("Depth Window", cv_image)
                cv2.waitKey(1)

        def shutdown(self):
        # stop turtlebot
```

```
                rospy.loginfo("Stop TurtleBot")
                rospy.loginfo("Stopping code -- Connnor")
                # a default Twist has linear.x of 0 and angular.z of 0.  So it'll stop TurtleBot
                #self.cmd_vel.publish(Twist())
                # sleep just makes sure TurtleBot receives the stop command prior to shutting
down the script
                rospy.sleep(1)

if __name__ == '__main__':
        try:
                DisplayData()
        except:
                rospy.loginfo("DisplayData node terminated.")
```