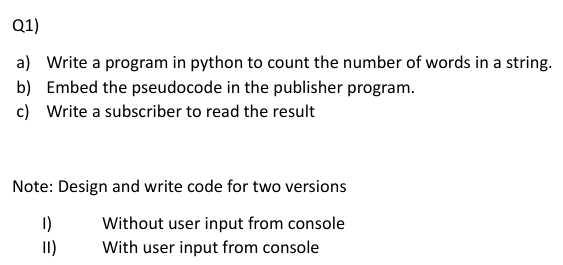
**-22AIE442-  
Robotics and operating systems**

**Name: Girish S Roll: AM.EN.U4AIE22044**

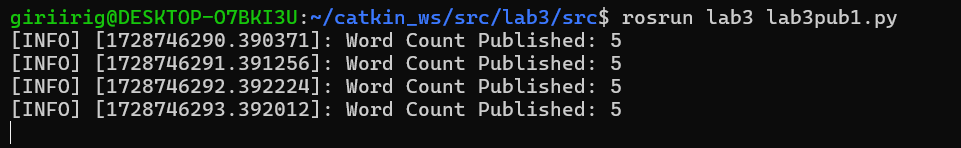
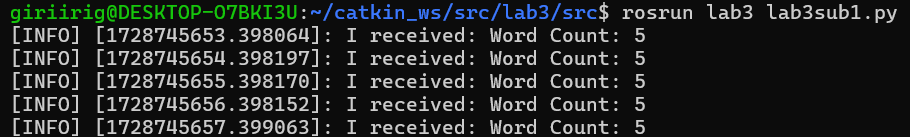
# **Labsheet 3**



## **I – No User Input**

#!/usr/bin/env python3  
import rospy  
from std\_msgs.msg import String  
def Publisher():  
    rospy.init\_node('publisher1', anonymous=True)  
    pub = rospy.Publisher('nogui', String, queue\_size=10)  
    rate = rospy.Rate(1)  
    string = "Hello world test test test".strip()  
    count = len(string.split())  
    while not rospy.is\_shutdown():  
        pub.publish(f'Word Count: {count}')  
 rospy.loginfo(f'Word Count Published: {count}')  
        rate.sleep()  
if \_\_name\_\_ == '\_\_main\_\_':  
    try:  
        Publisher()  
    except rospy.ROSInterruptException:  
        pass

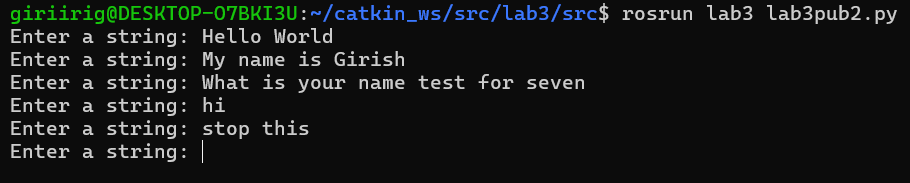
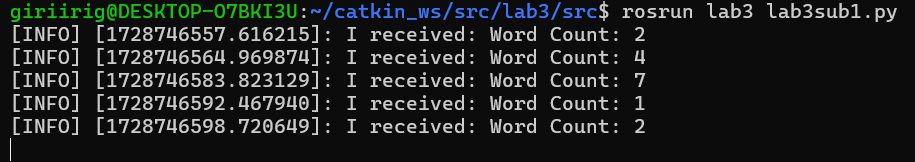
#!/usr/bin/env python3  
import rospy  
from std\_msgs.msg import String  
def callback(data):  
    rospy.loginfo(f"I received: {data.data}")  
def subscriber():  
    rospy.init\_node('subscriber1', anonymous=True)  
    rospy.Subscriber('nogui', String, callback)  
    rospy.spin()  
if \_\_name\_\_ == '\_\_main\_\_':  
    try:  
        subscriber()  
    except rospy.ROSInterruptException:  
        pass

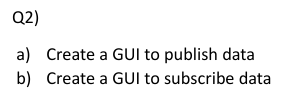


## **II – With User Input**

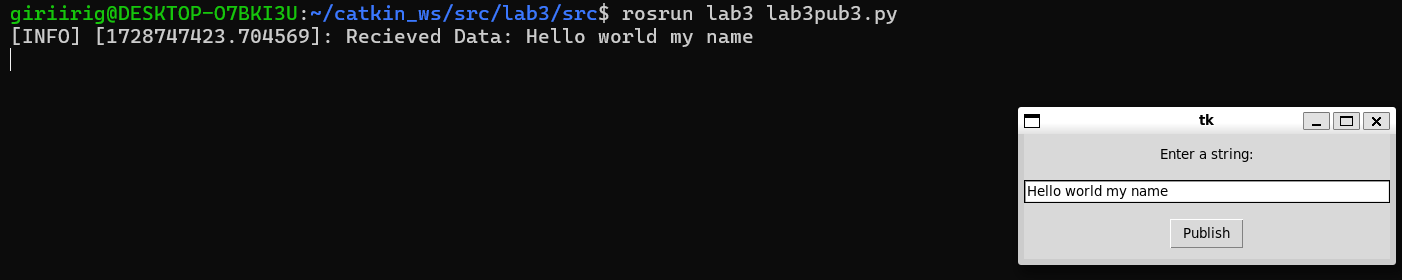
#!/usr/bin/env python3  
import rospy  
from std\_msgs.msg import String  
def Publisher():  
    rospy.init\_node('publisher2', anonymous=True)  
    pub = rospy.Publisher('nogui', String, queue\_size=10)  
    while not rospy.is\_shutdown():  
        string = input("Enter a string: ").strip()  
        count = len(string.split())  
        pub.publish(f'Word Count: {count}')  
if \_\_name\_\_ == '\_\_main\_\_':  
    try:  
        Publisher()  
    except rospy.ROSInterruptException:  
        pass

#!/usr/bin/env python3  
import rospy  
from std\_msgs.msg import String  
def callback(data):  
    rospy.loginfo(f"I received: {data.data}")  
def subscriber():  
    rospy.init\_node('subscriber1', anonymous=True)  
    rospy.Subscriber('nogui', String, callback)  
    rospy.spin()  
if \_\_name\_\_ == '\_\_main\_\_':  
    try:  
        subscriber()  
    except rospy.ROSInterruptException:  
        pass



## **I – Publisher with GUI**



#!/usr/bin/env python3  
import rospy  
from std\_msgs.msg import String  
import tkinter as tk  
def publish():  
    data = entry.get()  
    count = len(data.split())  
    pub.publish(f'Word Count: {count}')  
 rospy.loginfo(f'Recieved Data: {data}')  
def publisher():  
    global pub  
    rospy.init\_node('publisher3', anonymous=True)  
    pub = rospy.Publisher('withgui', String, queue\_size=10)  
    root = tk.Tk()  
    label = tk.Label(root, text="Enter a string:")  
    label.pack(pady=10)  
    entry = tk.Entry(root, width=40)  
    entry.pack(pady=5)  
    button = tk.Button(root, text="Publish", command=publish)  
    button.pack(pady=10)  
    root.mainloop()  
if \_\_name\_\_ == '\_\_main\_\_':  
    try:  
        publisher()  
    except rospy.ROSInterruptException:  
        pass

## **II – Subscriber with GUI**

#!/usr/bin/env python3  
import rospy  
from std\_msgs.msg import String  
import tkinter as tk  
from tkinter import messagebox  
def callback(data):  
    root.after(0, show\_message, f"Message: {data.data}")  
def show\_message(message):  
    messagebox.showinfo("Received Message", message)  
def subscriber():  
    global root  
    rospy.init\_node('subscriber2', anonymous=True)  
    rospy.Subscriber('withgui', String, callback)  
    root = tk.Tk()  
    root.protocol("WM\_DELETE\_WINDOW", on\_close)  
    root.mainloop()  
def on\_close():  
    rospy.signal\_shutdown("GUI closed")

    root.destroy()

if \_\_name\_\_ == '\_\_main\_\_':  
    try:  
        subscriber()  
    except rospy.ROSInterruptException:  
        pass

