

# **EECS/BioE 106A/206A**

# **Lab 1: Introduction to ROS**

## **(Turtleism!)**

Two people per lab station please!



# Welcome to Lab!

Health is the #1 priority.  
If you are not feeling well, please let us know



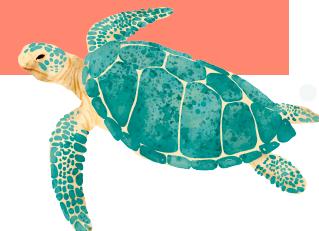
## Covid Safety Rules

- Be respectful to everyone
- Keep your stations clean
- No food/drink in the lab
- Don't work on the lab alone



## Introductions

Name, pronouns, major/year, and thing you're looking forward to in this class



# Lab Schedule



Lab 1

Lab 2

Module A  
Labs 3, 4 +  
Buffer Week

Module B  
Labs 5, 6 +  
Buffer Week

Module C  
Labs 7, 8 +  
Buffer Week



# Lab Philosophy

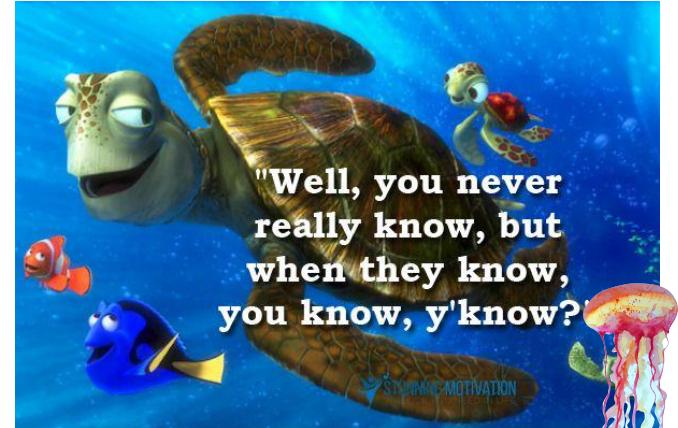
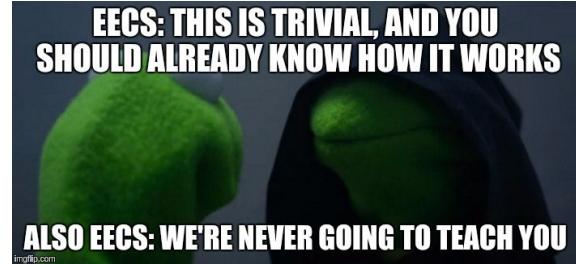
- Learn how to program real robots using the **Robotic Operating System** (ROS)
- Get good at debugging both **hardware** and **software**
- **Have fun** getting your hands dirty with labs
- Make **friends!** Robotic or Human
  - Google
  - StackoverFlow
  - Tutorials
  - Blogs, etc



<https://mashable.com/2015/08/24/baxter-robot-connect-four/>

# Don't Be Intimidated

- It's not magic, you got this!
- ... but just because it's not magic doesn't mean it's trivial
- You can't possibly know all of this already.
- Everyone is coming in with different kinds of expertise.



# Lab Structure



1

## Meeting

Beginning of Class

Go over material in the lab

Review FAQs



2

## Work on Lab

Work with a partner on the labs



3

## Help & Checkoff Queue

Fill out a request for the help / checkoff queue

# Lab Mechanics

- Form groups of 2 people
- Make Friends!





# Any Logistical Questions?

# Lab 1!



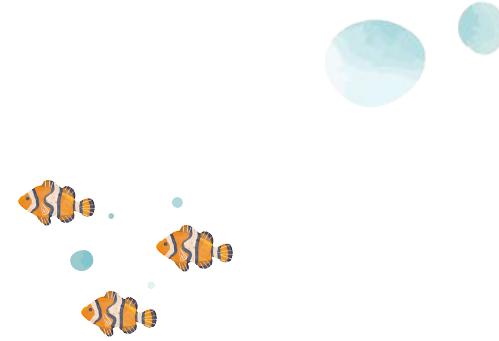
# Key Takeaways

01

ROS

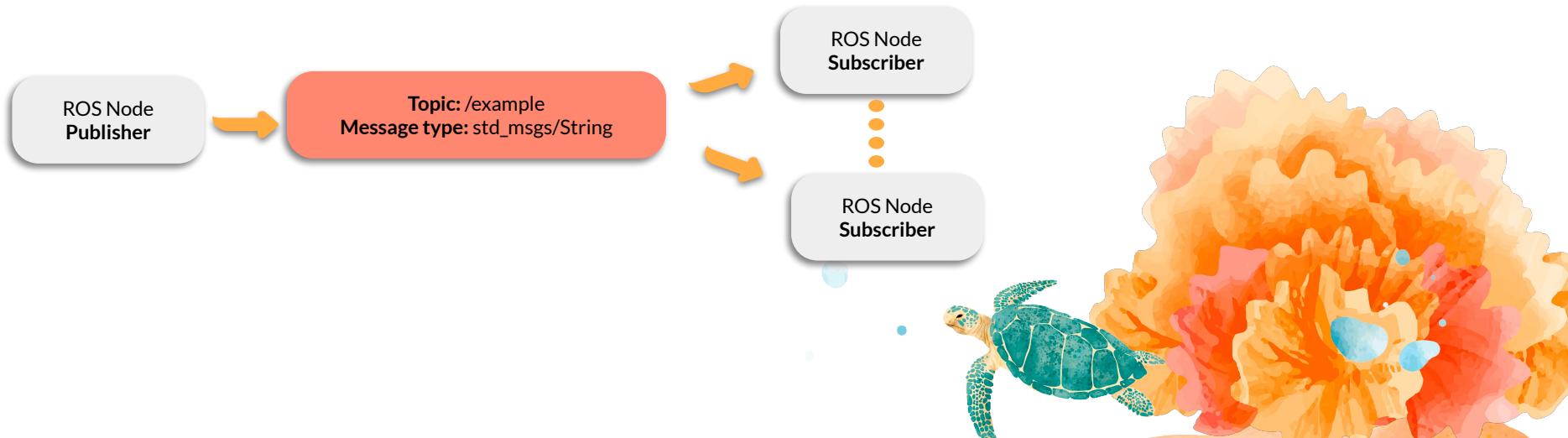
02

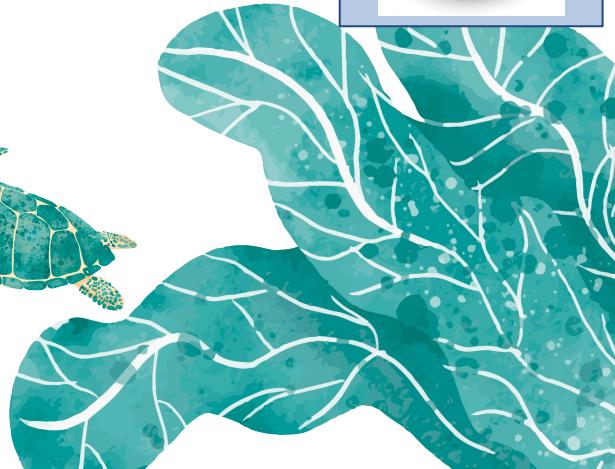
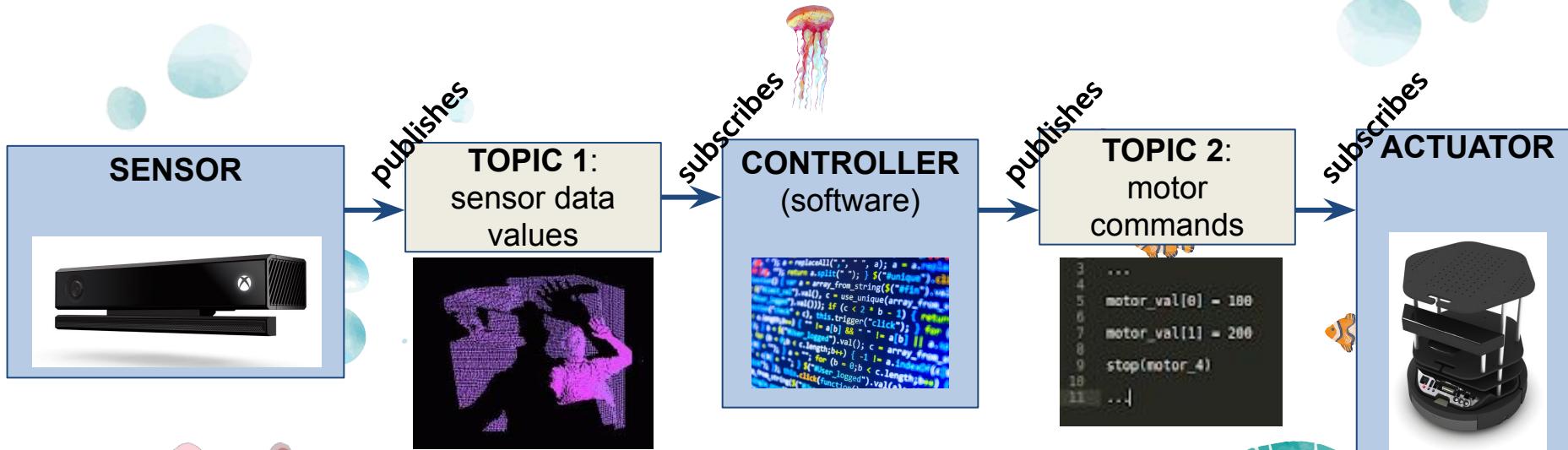
Lab Objectives



# ROS - not really an “OS”

- An open-source, cross-platform pseudo-operating system intended for distributed robotics applications
- Not really an “operating system,” just a series of libraries that allow hardware and sensors to talk to each other asynchronously or synchronously via event-driven programming
- All coordinated by a master node





# ROS



## Nodes

Processes that perform computation



## Topics

Queues over which nodes exchange messages



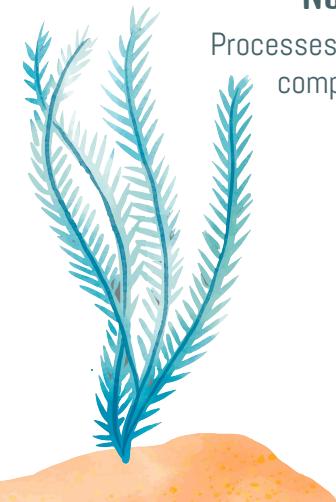
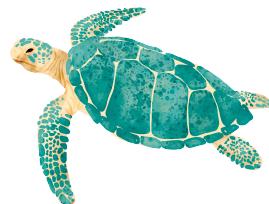
## Publisher

Node that sends message to a topic



## Subscriber

Node that receives message from a topic



# Key Takeaways

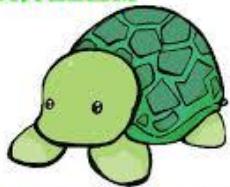


- Set up a new ROS environment, including creating a new workspace and creating a package with the appropriate dependencies specified
- Use the `catkin` tool to build the packages contained in a ROS workspace
- Run nodes using `rosrun`
- Use ROS's built-in tools to examine the topics and services used by a given node



# Tips

TURTLEISM



YOUR NOT ALONE...



Save your code on GitHub, privately

The Internet is your best friend

Help/Checkoff Queue:

<https://tinyurl.com/fa23-106alab>

Fun: How many Turtles did you count?



# THANKS!

Does anyone have any questions?

