1. **What are ROS Nodes?**

ROS nodes are individual processes that run code and communicate with other nodes via publishing and listening.

2. **What are ROS Topics?**

ROS Topics are buses which nodes use to exchange messages.

3. **What are ROS Workspaces?**

ROS workspaces are directories where ROS packages are managed.

4. **What are ROS services?**

ROS services define communication between the nodes, where one node requests a service from another, which performs a task and sends back a response.

5. **When would you use a ROS service vs a ROS topic?**

ROS topics are better for a continuous flow of data, such as telemetry from a sensor, but services are better for one-time requests.

6. **What do you have to do every time you open a new terminal in order to use ROS?**

You must source the setup file with source /opt/ros/humble/setup.bash or source install/setup.bash

7. **I have just completed writing a node in my_robot_controller in a workspace called ros2_ws. What does my path look like?**

~/ros2_ws/src/my_robot_controller/my_robot_controller/my_first_node.py

8. **If I were to have successfully established a ROS environment and run both talker and listener what would running rqt_graph produce?**

rqt_graph would display a diagram showing the talker node publishing to a topic and the listener node subscribing to that topic.

9. **When you create a new node what do you need to do in order to run a new node called tester found in the node_tester package and where would you run it?**

In the workspace folder (ros2_ws, for instance) run colcon build and then ros2 run node_tester tester.

10. **What do you need to source to run custom nodes?**

We need to source the workspace with source ~/ros2_ws/install/setup.bash, or just source ~/.bashrc once if the previous line is pasted to the end of the file.

11. **If I have created a node called test_node in my_robot_controller and would like to execute it through the command line. How would I make it executable from the command line with the ros2 functionalities? Name it tester.**

Add "if __name__ == '__main__': main()" to the node and "test_node = my_robot_controller.my_first_node:main" under setup.py, source the setup file, and make it executable with chmod +x test_node.py.

12. What packages do you need to import for every node?

Import rclpy and from rclpy.node import Node

13. What are the arguments for ros publisher and subscriber?

Publishers' arguments are their topic name, message type, and (optionally) their queue size. Subscribers' arguments are their topic name, message type, and callback function.

14. What does ros spin do and why do you need it?

ros::spin() keeps your node active, allowing it to process callbacks and remain responsive.

15. What is a callback?

A callback is a function in a node that gets called whenever a message is received on a subscribed topic.

16. How do I see the ROS Topics running?

Use the command ros2 topic list.

17. I noticed there is a topic called geometry message. How can I see what information is on that topic?

Use the command ros2 topic echo /geometry_message.

18. Once I know the name of a topic how do I know the message type of it?

Use ros2 topic info /<topic name>.

19. What is the first thing you should do if you run into an error?

Make sure all the necessary setup files are sourced, Google the error, or ask your tablemates/TA/Immanuel (in that order).

20. In setup.py I add the line “test_node = my_robot_controller.my_first_node ” what is the executable name, what is the package name, and what is my node name?

The executable name is test_node, the package name is my_robot_controller, and the node name is my_first_node.

21. How do you edit a python file in the terminal?

Use a command-line text editor by running nano <filename>.py.

22. What does chmod +x do?

It makes a file executable by setting the execute permission on the file.

23. What is a src folder and why is it necessary?

The src folder contains the source code for the ROS packages, including the code for all the nodes. It keeps your project organized so nodes can easily be located and modified.

24. How do you create a ros package?

Use the command ros2 pkg create my_robot_controller --build-type ament_python --dependencies rclpy.

25. Why should you include --symlink in colcon build?

Using --symlink links files instead of copying them, making it easier to test changes immediately because if you change your source code, you don't have to compile them again.