1. **What are ROS Nodes?**
   a. A ROS node is a part of a communication network that can perform computation and exchange information with other nodes.
2. **What are ROS Topics?**
   a. ROS topics are devices that transmit information between nodes. Nodes can publish or subscribe to an individual topic.
3. **What are ROS Workspaces?**
   a. A ROS workspace is a directory that has ROS packages
4. **What are ROS services?**
   a. Services are devices that transmit information but also reply to the original "publisher" of information.
5. **When would you use a ROS service vs a ROS topic?**
   a. You use a ROS topic when you only want to send one-way information that can have multiple publishers/subscribers. You use a service when you want a response back. Services also should not be continuously run.
6. **What do you have to do every time you open a new terminal in order to use ROS?Hint: it will produce the error ROS2 command not found. Bonus: How do you get around this?**
   a. You have to use the source /opt/ros/humble/setup.bash. To get around this, add it to the bottom of your bashrc file
7. **I have just completed writing a node in my_robot_controller in a workspace called ros2_ws what does my path look like?**
   a. ~/ros2_ws/src/my_robot_controller/my_robot_controller/node_is_here.py
8. **If I were to have successfully established a ROS environment and run both talker and listener what would running rgt_graph produce?**
   a. (talker) ->[/chatter]->(listener)
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?**
   a. You would need to source setup.bash, and run it using ros2 run pkg_name executable_name
10. **What do you need to source to run custom nodes?**
    a. source ~/ros2_ws/install/setup.bash
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. Hint: you must add something to a .py file.**
    a. Add another object to the console scripts list in setup.py with the executor, package name, and install name.
12. **What packages do you need to import for every node?**
    a. rclpy and rclpy.node
13. **What are the arguments for ros publisher and subscriber?**
    a. Message type, topic name, queue size, callback function for subscriber
14. **What does ros spin do and why do you need it?**
    a. Ros spin ensures that the node is kept alive until it is killed by the user

15. **What is a call back?**
    a. A function that is passed as an argument and executed under a certain condition
16. **How do I see the ROS Topics running?**
    a. ros2 topic list
17. **I noticed there is a topic called geometry message. How can I see what information is on that topic?**
    a. ros2 topic echo /geometry_message
18. **Once I know the name of a topic how do I know the message type of it?**
    a. ros2 topic info /geometry_message
19. **What is the first thing you should do if you run into an error?**
    a. DON'T PANIC👍
20. **In setup.py I add the line "test_node = my_robot_controller.my_first_node:main" what is the executable name, what is the package name, and what is my node name?**
    a. test_node: executable name
    b. my_robot_controller: package name
    c. my_first_node: node name
21. **How do you edit a python file in the terminal?**
    a. vim file.py
22. **What does chmod +x do?**
    a. It makes the python file executable
23. **What is a src folder and why is it necessary?**
    a. It houses all of your python files with nodes, it has package.xml, and setup.py
24. **How do you create a ros package?**
    a. ros2 pkg create pkg_name --build-type ament_python --dependencies rclpy
25. **Why should you include --symlink in colcon build?**
    a. This ensures that all updates are immediately accounted for, and you don't have to build it after every change you make to your python file
26. **Bonus: Create a Ros node that will log : " UAV is Awesome"?**

In: ros2_ws/src/my_robot_controller/my_robot_controller/uav_awesome.py

```
#!/usr/bin/env python3
import rclpy
from rclpy.node import Node

class MyNode(Node):
    def __init__(self):
        super().__init__("first_node")
        self.counter_ = 0
        self.get_logger().info("UAV is Awesome")

def main(args=None):
    rclpy.init(args = args)
    node = MyNode()
    rclpy.spin(node)
    rclpy.shutdown()
if __name__ == "__main__":
    main()
```