src/bot/include/CPose.h

```cpp
// This header file is for class CPosewhich has the capability to
// calculate position from odom msgs
// It publishes pose to Drive class . The purpose of this is to isolate
// capability of reading in odom values and calculate position value so
// that errors can be identified easily -- Follow OOP design
#ifndef CPOSE_H_
#define CPOSE_H_
#include <ros/ros.h>
#include <sensor_msgs/LaserScan.h>
#include <geometry_msgs/Twist.h>
#include <nav_msgs/Odometry.h>
#include <vector>
#include<std_msgs/Float64.h>


const char topicName[] = "POSE";
// Set queue size big to prevent loss if any
// delay occurs
//https://stackoverflow.com/questions/56444248/reason-to-set-queue-size-of-ros-
publisher-or-subscriver-to-a-large-value
const int QSize = 1000;

/// @brief---
// CPose interface------------------------------------------------------
// This class subscribes to Odometry and acquire pose of the bot. It then
// publish turtlebot pose to Drive class for motion planning. It also stores
// current linear and angular velocity for other uses (not yet identified,
// but kept for debugging).
class CPose{
  public:
    CPose();
    ~CPose();
    void odomMsgCallBack(const nav_msgs::Odometry::ConstPtr &msg);
    void PublishPose();

  private:
    // ROS NodeHandle
    ros::NodeHandle nh_;
    ros::NodeHandle nh_priv_;

    // Subscriber to odometry
    ros::Subscriber odomSub;

    // Publisher
    ros::Publisher botPub;

    // Pose data from odometry
    double curLinVel;
    double curAngVel;

    double tb3Pose;
    std_msgs::Float64 msg;
};

#endif
```