

Peer Code Review of Assessment Task 2

Following Assessment Tasks 1, 2 and 3 you will be required to complete a peer code review of assignment submissions by your classmates. This document gives some guidance on how to assess your peers work on the assessment criteria provided. This document also gives the criteria by which your peer code review will be assessed.

Guidelines for Applying Assessment Task 2 Criteria

Encapsulation of data and appropriate access methods:

Ranger classes (radar, sonar, laser) should:

- Inherit from common base class
- Keep member data private/protected
- Have a constructor that sets valid default values for sensor parameters
- Provide public method(s) to query/set parameters specific to this sensor type
- Validate any configurable values supplied as arguments to setters and use valid defaults otherwise
- Indicate to the calling code whether the values supplied as arguments to setters are valid (eg. return a bool)
- Avoid duplicating/overriding implementation of methods common to all sensors

Common ranger base class should:

- Be abstract (contain at least one pure virtual method)
- Contain (protected) member variables common to all sensor types
- Implement methods common to all sensor types
- Provide public method(s) to retrieve data from the sensor
- Provide public method(s) to query/set parameters common to all sensor types

Proper code execution

The program should:

- Compile and run successfully
- Display the fixed parameters of the sensor
- Generate sensor data as per description (number of samples and fov)
- Allow the user to specify values for the configurable parameters
- Not allow the user to specify invalid values
- Produce fused range output as a result of combining output from the three rangers consistent with the behaviour specified in the documentation
- Appropriately select the max, min, or mean range value per angle according to the fusion method selected
- Handle sensor boundary values with justifiable logic specified in documentation, not simply disregard them.

Documentation

The documentation should:

- Be automatically generated from mark-up comments in the source code
- Include descriptions of all classes
- Include descriptions of all methods
- Include descriptions of all input parameters
- Include descriptions of all return values
- Important elements of program logic are explained

Modularity

Ranger classes (radar, sonar, laser) should:

- Each have a separate header file (*.h OR *.hpp) for the class declaration
- Each have a *.cpp file for the class definition, separate to the header and to other classes
- Implement all functionality specific to the sensor internally (eg. delay when polled for data)
- Not depend on the data fusion class in any way (eg. Should NOT #include data fusion's source)
- Depend on the ranger base class

Common ranger base class should:

- Have a separate header file (*.h OR *.hpp) for the class declaration.
- Have a *.cpp file for the class definition, separate to the header and to other classes.
- Implement any functionality common to all ranger sensors internally (eg. delay when polled for data)
- Not depend on the data fusion class in any way

Data fusion class should:

- Depend on ONLY the common ranger base class
- NOT depend on the ranger sub-classes (radar, sonar, laser) in any way
- Deal with all rangers passed to it equivalently and agnostically. I.e. the **type**, **order** and **number** of rangers provided should not adversely affect the operation of the data fusion class

Assessment Criteria for the Peer Code Review – Assessment Task 4 – Part 2 of 3

Assessment Task 4, Peer Code Review has a weighting of 15% in the overall assessment of this course. Your peer code review of Assessment Task 2 will contribute to 3 of these 9 points. The Assessment criteria is explained below:

1. Evaluates the submissions, broadly identifies submission correctness as per criteria. (2 points)
2. Provides feedback on improving submission, 0.33 point for meaningful feedback per submission. (1 point)
3. Reflects on own work and identifies potential improvements from reviewed submissions or correctly justifies why own work need not be improved from interaction with reviewed submissions (0 points)