**Experiment Report - 66 - test3\_demoCode**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | line 9 | Semantic | √ | √ | √ |
| E02 | line 19 | Semantic | × | × | × |
| E03 | line 28 | Semantic | √ | × | × |

Additional Errors Found by Self: 0

Self-Review Detection Rate: 67%

Peer 1 Detection Rate: 33%

Peer 2 Detection Rate: 33%

1. **Source Code**
2. #include <string>
3. #include <fstream>
4. void VideoRecorder::writeMetadataHeader() {
5. videoFileStream << "Metadata: \n";
6. videoFileStream << "FrameRate: " << videoConfig.getFrameRate() << " fps\n";
7. videoFileStream << "Resolution: " << videoConfig.getResolution() << "\n";
8. videoFileStream << "Encoding: " << videoConfig.getEncodingStandard() << "\n";
9. videoFileStream << "Bitrate: " << videoConfig.getBitrate() << " kbps";
10. videoFileStream << "Timestamp: " << timeManager.getCurrentTimestamp() << "\n";
11. std::cout << "Metadata header written to video file." << std::endl;
12. }
13. void VideoRecorder::embedTimestampAndGPS() {
14. std::string timestamp = timeManager.getCurrentTimestamp();
15. std::pair<double, double> gpsCoordinates = gpsModule.getCoordinates();
16. videoFileStream << "Timestamp: " << timestamp << "\n";
17. videoFileStream << "GPS Coordinates: [" << gpsCoordinates.first << " " << gpsCoordinates.second << "]\n";
18. }
19. void VideoRecorder::setupRecording() {
20. int frameRate = videoConfig.getFrameRate();
21. std::string resolution = videoConfig.getResolution();
22. std::string encodingStandard = videoConfig.getEncodingStandard();
23. int bitrate = videoConfig.getBitrate();
24. std::cerr << "Setting up recording with Frame Rate: " << frameRate
25. << ", Resolution: " << resolution
26. << ", Encoding: " << encodingStandard
27. << ", Bitrate: " << bitrate << " kbps" << std::endl;
28. }