**Experiment Report - 60 - test7\_OnlineExamSystem**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | line 38 | Semantic | √ | √ | √ |
| E02 | line 48 | Syntax | × | × | √ |
| E03 | line 55 | Semantic | √ | √ | √ |
| E04 | line 70 | Semantic | √ | × | × |

Additional Errors Found by Self: 0

Self-Review Detection Rate: 75%

Peer 1 Detection Rate: 50%

Peer 2 Detection Rate: 75%

1. **Source Code**
2. package a;
3. import java.util.\*;
4. class Student {
5. private String name;
6. private double score;
7. public Student(String name, double score) {
8. this.name = name;
9. this.score = score;
10. }
11. public String getName() {
12. return name;
13. }
14. public double getScore() {
15. return score;
16. }
17. }
18. class ExamSystem {
19. private List<Student> students;
20. private List<String> rankingHistory;
21. public ExamSystem() {
22. this.students = new ArrayList<>();
23. this.rankingHistory = new ArrayList<>();
24. }
25. public boolean recordScore(String name, double score) {
26. if (score >= 0 && score <= 100) {
27. students.add(new Student(name, score));
28. rankingHistory.add(String.format("Recorded: %s - %.1f", name, score));
29. System.out.printf("Score recorded: %s - %.1f%n", name, score);
30. return true;
31. } else {
32. System.out.println("Invalid score! Must be between 0 and 10.");
33. return false;
34. }
35. }
36. public void rankStudents() {
37. students.sort((s1, s2) -> Double.compare(s2.getScore(), s1.getScore()));
38. System.out.println("\nExam Rankings:");
39. for (int i = 0; i < students.size(); i++) {
40. String distinction = (students.get(i).getScore() > 90) ? " (Distinction)" : ;
41. System.out.printf("%d. %s - %.1f%s%n", i + 1, students.get(i).getName(), students.get(i).getScore(), distinction);
42. rankingHistory.add(String.format("Ranked: %d. %s - %.1f%s", i + 1, students.get(i).getName(), students.get(i).getScore(), distinction));
43. }
44. }
45. public void printRankingHistory() {
46. System.out.println("Ranking History:");
47. for (String log : rankingHistory) {
48. System.out.println(log);
49. }
50. }
51. }
52. public class c10\_OnlineExamSystem {
53. public static void main(String[] args) {
54. ExamSystem exam = new ExamSystem();
55. // testcase-VT:
56. exam.recordScore("Alice", -1.53); // score < 0
57. exam.recordScore("Bob", 2.78); // score >= 0 && score <= 90
58. exam.recordScore("Charlie", 92.64); // score > 90 && score <= 100
59. exam.recordScore("Alice", 104.22); // score > 100
60. // testcase-FT:
61. // exam.recordScore("Alice", 7.022238808055921E305);
62. // exam.recordScore("Bob", 1.7976931348623157E308);
63. // exam.recordScore("Charlie", 0.0);
64. // exam.recordScore("David", -2.2250738585072014E-308);
66. exam.rankStudents();
67. exam.printRankingHistory();
68. }
69. }