

Tasks

Tasks	Stories	Related Documents/Notes
<input checked="" type="checkbox"/> 1 Settle on an approach for handling missing values	S1-S9	Use Double objects rather than double values and use null to represent missing values
<input checked="" type="checkbox"/> 2 Design a utility class for working with missing elements	S1-S9	The specifications for the Missing class
<input checked="" type="checkbox"/> 3 Design unit tests for the class in task 2	S1-S9	The description of the unit tests for the Missing class
<input type="checkbox"/> 4 Implement the unit tests in task 3	S1-S9	
<input type="checkbox"/> 5 Implement the class in task 2	S1-S9	
<input type="checkbox"/> 6 Test (and debug if necessary) the class in task 8	S1-S9	
<input checked="" type="checkbox"/> 7 Design an approach for representing non-missing scores that can be associated with a weight	S1-S9	UML diagram for the Score class and specification 1 of the Score class
<input checked="" type="checkbox"/> 8 Design an approach for representing missing scores that can be associated with a weight	S1-S9	Specifications 2-4 of the Score class
<input type="checkbox"/> 9 Design unit tests for the class in tasks 7-8	S1-S9	
<input type="checkbox"/> 10 Implement the unit tests for the class in task 7-8	S1-S9	
<input type="checkbox"/> 11 Test and debug the class in task 7-8	S1-S9	
<input checked="" type="checkbox"/> 12 Determine how to handle size issues		The specifications for the SizeException class
<input checked="" type="checkbox"/> 13 Design a class that can be used to drop the highest and lowest scores	S1	The specifications for the DropRule class
<input checked="" type="checkbox"/> 14 Determine whether the class in task 13 should implement an interface	S1	The Rule interface in the design document
<input checked="" type="checkbox"/> 15 Design unit tests for the class in task 13	S1	The description of the unit tests for

the DropRule class

<input type="checkbox"/> 16	Implement the unit tests for the class in task 13	S1	
<input type="checkbox"/> 17	Implement the class in task 13	S1	
<input type="checkbox"/> 18	Test (and debug, if necessary) the class in task 13	S1	
<input checked="" type="checkbox"/> 19	Design a class that can be used to calculate weighted totals	S1	The specifications for the WeightedTotalSystem class
<input checked="" type="checkbox"/> 20	Determine whether the class in task 19 should implement an interface	S1	The ScoringSystem interface in the design document
<input checked="" type="checkbox"/> 21	Design a class that can be used to calculate totals	S1	The specifications for the TotalSystem class
<input type="checkbox"/> 22	Determine the relationship between the classes in tasks 19 and 21		
<input checked="" type="checkbox"/> 23	Design one integration test with complete information for the class in task 19	S3	Tests: IT Complete 01
<input checked="" type="checkbox"/> 24	Design integration test with missing weights for the class in task 19	S4	Tests: IT Missing Weight 01, IT Missing Weight 02
<input checked="" type="checkbox"/> 25	Design one integration test with invalid weights for the class in task 19	S5	Tests: IT Invalid Weight 01
<input checked="" type="checkbox"/> 26	Design integration tests with missing scores for the class in task 19	S6	Tests: IT Missing Score 01, IT Missing Score 02, IT Missing Score and Missing Weight 01
<input type="checkbox"/> 27	Implement the integration tests for the class in task 19		
<input type="checkbox"/> 28	Design integration tests for the class in task 21		
<input type="checkbox"/> 29	Implement the integration tests for the class in task 21		
<input type="checkbox"/> 30	Implement the classes in tasks 19 and 21		
<input type="checkbox"/> 31	Test (and debug if necessary) the classes in tasks 19 and 21		
<input checked="" type="checkbox"/> 32	Design an approach for representing a diver's position	S9	The specifications for the Position enum
<input type="checkbox"/> 33	Implement the class/enum in task 32		
<input type="checkbox"/> 33	Test (and debug if necessary) the class/enum in tasks 32 and 33		

<input checked="" type="checkbox"/> 34 Implement the code that accesses the command line arguments	S7	IndividualScorer.java
<input checked="" type="checkbox"/> 35 Implement the code that converts "N/A" arguments to missing values	S8	IndividualScorer.java
<input checked="" type="checkbox"/> 36 Implement the code that calculates and displays the scores	S9	IndividualScorer.java
<input checked="" type="checkbox"/> 37 Create two system tests with no missing scores	S9	Tests: ST Complete 01, ST Complete 02
<input checked="" type="checkbox"/> 38 Create test with one missing score for one judge	S9	Test: ST Missing 01
<input type="checkbox"/> 39 Create Eclipse "Run Configurations" for each test	S7, S8, S9	
<input type="checkbox"/> 40 Test the system	S7, S8, S9	
<input type="checkbox"/> 41 Debug the system (if necessary)	S1-S9	