Tasks

Tasks		Stories	Related Documents/Notes
☑1	Identify the signature for a method for calculating a weighted total	S1	
☑2	Organize the method in task 1 into a class	S1	The specifications for the `WeightedTotalStrategy` class
☑3	Determine how to handle missing weights	S1	Specification 2.4.1 for the `WeightedTotalStrategy` class
☐4	Create unit tests for the class in task 2	S1	
☐5	Test and debug the class in task 2	S1	
☑6	Identify the signature for a method for dropping the lowest grade	S1	
☑7	Organize the method in task 6 into a class	S1	The specifications for the `DropFilter` class
☑8	Determine how to handle size issues	S1	Specification 4 for the `DropFilter` class and the specifications for the `SizeException` class
☐9	Create unit tests for the class in task 7	S1	
☐10	Test and debug the class in task 7	S1	
☑11	Identify the signature for a method for calculating a total	S2	
☑12	Organize the method in task 1 into a class	S2	The specifications for the `TotalStrategy` class
☑13	Determine how to handle missing weights	S2	Specification 2.4.1 for the `TotalStrategy` class

☐14	Create unit tests for the class in task 12	S2	
☐15	Test and debug the class in task 12	S2	
☑16	Design an approach for representing non-missing values that can be associated with a weight	S3	UML diagram for the `Grade` class and specification 1 of the `Grade` class
☑17	Design an approach for representing missing values that can be associated with a weight	S3	Specifications 2-4 of the `Grade` class
☐18	Create unit tests for the class in tasks 16-17	S3	
☐19	Test and debug the class in task 16-17	S3	
☑20	Design an approach for manipulating missing values numerically	S3	UML diagram and specifications for the `Missing` class
☐21	Create unit tests for the class in task 20	S3	
☐22	Test and debug the class in task 20	S3	
☑23	Organize all of the classes in the system		UML class diagram of the system
☑24	Implement the code that accesses the command line arguments	S4	
☑25	Implement the code that converts “NA” arguments to missing values	S5	
☑26	Implement the code that calculates and displays the course grade	S6	
☑27	Create three tests with no missing values	S1, S2, S4, S6	Tests: Complete 01, Complete 02, Complete 03
☑28	Create test with one missing value in one category	S3, S5, S6	Test: Missing One in One Category
☑29	Create test with one missing value in each category	S3, S5, S6	Test: Missing One in Each Category
☑30	Create test with multiple missing values in each category	S3, S5, S6	Test: Missing Multiple in Each Category

☑31	Create test with all missing values	S3, S5, S6	Test: Missing All
☐32	Create Eclipse “Run Configurations” for each test	S1-S6	
☐33	Test the system	S1-S6	Note: Since the Gradient class has already been completed, integration testing will not be conducted. Instead, the system tests will be used for both purposes.
☐34	Debug the system (if necessary)	S1-S6	