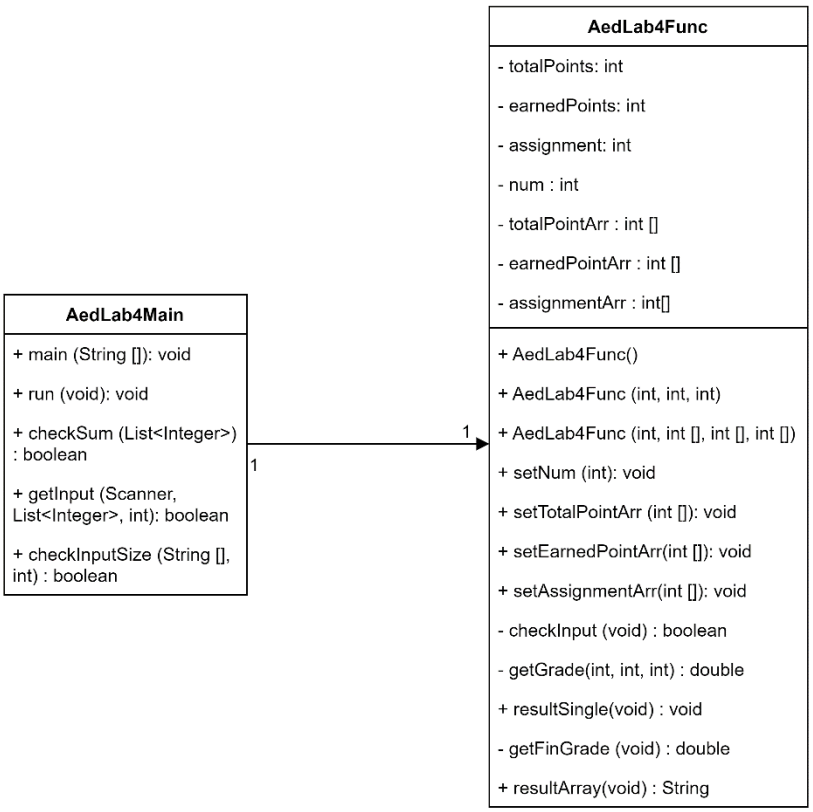

Above image is the UML class diagram for the weighted grading application.
The application has 2 main class: AedLab4Main and AedLab4Func.
The relationship between these 2 classes is 1-to-1 Association, AedLab4Main knows AedLab4Func, initial it and use its functions.

Methods in the AedLab4Main:

Name & visible	public main
Parameter	String []
Return	void
Description	The application starts from this function and also holds the exception throw.

Name & visible	public run
Parameter	void
Return	void
Description	The main function in this class, initialize the scanner and buffers for user typing and call the result function.

Name & visible	public checkSum
Parameter	List<Integer> list
Return	boolean
Description	Get the assignment percentage list and check whether its sum is 100. Return true if yes.

Name & visible	public getInput
Parameter	Scanner sc, List<Integer> buffer, int N
Return	boolean
Description	Get the input scanner, buffer list and the desired number of inputs. Read the input from Scanner as String, and split it into list, meanwhile checking the number of inputs is correct. Return true if number of input is correct and loading input success.

Name & visible	public checkInputSize
Parameter	String [] array, int N
Return	boolean
Description	Get the input string array and desired number of input. Return true if length of string array is equal to desired number of input.

Attributes in the AedLab4Func:

Name & visible	Type	Description
private totalPoints	int	the total point of one assignment (for Assign1)
private earnedPoints	int	the earned point of one assignment (for Assign1)
private assignment	int	the assignment percentage (for Assign1)
private num	int	Number of assignments
private totalPointArr	int[]	array of total point for multi-assignments
private earnedPointArr	int[]	array of earned point for multi-assignment
private assignmentArr	int[]	The array of assignments percentage

Functions in the AedLab4Func:

Name & visible	public AedLab4Func
Parameter	void
Return	void
Description	basic constructor

Name & visible	public AedLab4Func
Parameter	int t, int e, int a
Return	void
Description	constructor accept the 3 int for totalPoint, earnedPoint and assignment. (for Assign1)

Name & visible	public AedLab4Func
Parameter	int n, int[] t, int[] e, int[] a
Return	void
Description	constructor accept the number of assignments and 3 int arrays for total points, earned point and assignments.

Name & visible	public setNum
Parameter	int num
Return	void
Description	Set the private attribute num

Name & visible	public setTotalPointArr
Parameter	int[] totalPointArr
Return	void
Description	Set the private attribute totalPointArr

Name & visible	public setEarnedPointArr
Parameter	int[] earnedPointArr
Return	void
Description	Set the private attribute earnedPointArr

Name & visible	public setAssignmentArr
Parameter	int[] assignmentArr
Return	void
Description	Set the private attribute assignmentArr

Name & visible	private checkInput
Parameter	void
Return	boolean
Description	Check whether the 3 attributes for assign 1 is valid. Return true if valid.

Name & visible	private getGrade
Parameter	int e, int t, int a
Return	double
Description	Get the weighted grade based on e / t * a.

Name & visible	public resultSingle
Parameter	void
Return	void
Description	Get the result for single grade (for Assign 1).

Name & visible	private getFinGrade
Parameter	void
Return	double
Description	Get the final weighted grade for multi-assignments

Name & visible	public resultArray
Parameter	void
Return	String
Description	Get the result for multi-assignments grades. Return the final level String.

Discussion of the limitations:

1. The first limitation is that although we create global exception catch at the main function, there is only 2 customized alert prompts:
   “Invalid Input Error:: number of input is different.”
   “Invalid Input Error:: weights sum is not 100”
   The rest errors are handled in raw prompts.

2. The error tolerance is not acceptable, once there is invalid input, user has to re-type all the information.

3. Although this application currently is simple and single thread program, maybe in the future, the AedLab4Func is running on the cloud and service need to be synchronized.

4. The input is String, split by comma. But after processing the type that inputs stored is int, other type of input will lead to exception prompt.

5. User Experience is not logic direct and straightforward, which still need extra information to help input.

Improvement for Corresponding limitations:

1. Handle more detailed different exceptions in each function, rather than directly throw them all from main function, to improve user experience and let user know the actual problem.

2. No longer directly exit the application when errored, but add the while(true) and check function to each input, once input is not valid, loop until it’s legal then break the infinite loop.

3. Change the array attributes in AedLab4Func to BlockedList or manually add Object synchronized lock to ensure the atomic change.

4. Better to change the stored data type from int to Double/double, so that the application can handle more complex points like decimals.

5. Combined with some graphics libraries like awt or swing making a GUI for users, so that the input format can be more straightforward.