# Specifications

Create a program that allows for the user of the program to enter their grades and save them in an array of doubles. The program should perform the following steps:

1. Ask the user for the number of grades to enter
2. Create a new array using the specified size from step 1
3. Ask the user for each of the grades to enter and store them in the array
4. Print the contents of the array to the console
5. Ask the user for an additional number of items to add to the array
6. Since arrays are of a fixed size, you will need to create a new array with the larger size and copy the values from the previous array
7. Ask the user to enter grades to fill the remaining space in the array
8. Print the contents of the array to the console
9. Ask the user for a position in the array to remove the value (you can replace it with a zero), and then do so
10. Print the contents of the array to the console
11. Ask the user for a position in the array to overwrite the value, then overwrite the value at that position with the value 1000
12. Print the contents of the array to the console

Next, recreate the program, but now using a List of doubles. The program should perform the following steps:

1. Create a new list
2. Ask the user to enter grades until they choose to stop
3. Print the contents of the list to the console
4. Ask the user for another grade to add to the list and add it; remember that lists can expand in size, so there is no need to create a new list
5. Print the contents of the list to the console
6. Ask the user for a position in the list to remove a value, and do so
7. Print the contents of the list to the console
8. Ask the user for a position in the list to overwrite a value, and replace that value with the value 1000
9. Print the contents of the list to the console
10. Ask the user for a position in the array to insert a new value, and insert the value 2000 there
11. Print the contents of the array to the console

These two versions of the program can both be created within the Driver file for the project. Create each in their own method (named ArrayGrades and ListGrades) and call each in the main method.

# Documentation

Make sure **all** code files are [**fully documented**](http://csciwww.etsu.edu/bailes/courses/1260/LectureMaterial/Policies%20Regarding%20Code%20Documentation.docx)– see the [**course documentation policies**](http://csciwww.etsu.edu/bailes/Courses/1260/LectureMaterial/Policies%20Regarding%20Code%20Documentation.docx) posted on D2L for the conventions and requirements including examples of proper documentation. Proper documentation includes following the naming conventions for classes, files, methods, variables, constants, and other identifiers.

# Submission

Your submission should be a single **.zip** file with a name in the format of ***1260-LastFirst-Lab3*** containing your entire VS project. Submit the **one** **zipped** **file** to the **Lab 3 Dropbox** on D2L.

# Sample Output

Text

Description automatically generated

Text

Description automatically generated