To complete these four tasks, you must follow the following guidelines.

1. You should build one class called DataSample that represents each data sample.

* Inside this class, you should have the following members:
  + **private** **int** label; (represent the label of this sample).
  + **private** **int** numOfAttributes; (represent the number of attributes for each data sample).
  + **private** **double**[] atrributes; (represent the attributes in an array).
* Inside this class, you should have the following methods:
  + **public** DataSample(**int** lb, **double**[] atr); (Constructor for this class. You should set label, numOfAttributes and atrributes for this sample inside this method).
  + **public** **void** setLabel(**int** lb); (Set the label of this sample).
  + **public** **int** getLabel(); (Get the label of this sample).
  + **public** **int** getnumOfAttributes(); (Get the number of attributes of this sample).
  + **public** **double**[] getAttributes(); (Get the attributes of this sample).
  + **public** **double** distance(DataSample dat); (Get the distance of this sample to another sample dat. We calculate the Euclidean distance between two samples. E.g.  and  are two samples. The distance between them is: .)

***Hint:*** You may call Math.pow(x,0.5) to get .

You should build one class called DataSet that represents the whole dataset obtained from either “wine.txt” or “testwine.txt”.

* Inside this class, you should have the following members:
  + **private** **DataSample[]** dataArray; (represent the array of data samples obtained from either “wine.txt” or “testwine.txt”).
* Inside this class, you should have the following methods:
  + **public** DataSet(String fileName); (Constructor for this class.)

***Hint***:

* String fileName is the filename that contains the data, such as “wine.txt” or “testwine.txt”.
* You should call getDataSetSize(String fileName) to get the number of samples in the data file, and then allocate the memory space for dataArray.
* You may use Scanner to read the data from the file.
* You may use the split() function to get the string array which contains the strings for the label and the other 13 attributes.
* You may use Integer.parseInt() to get the class label, which is an integer in the data file.
* You may use Double.parseDouble() to get the other 13 attributes.
* You may use DataSample(label,arrayAttribute); to construct the sample.
* Remember to close the file after processing.
  + **public** **int** getDataSetSize(String fileName); (Get the number of samples in the data file.)

***Hint***: You can use Scanner to find the number of lines inside the data file, (One line represents one sample.) Scanner.hasNextLine() to determine whether reach to the end of the file. You should handle the exception when the input file name is not found, and print an error message.

* + **public** **double**[] getMean(**int** label); (Get the class mean for class label).

***Hint***: You need to calculate the sample mean for class label.

* You should loop through all the attributes, and calculate the mean for each attribute.
* You should have an inner loop to loop through all samples. For each sample, you should check whether it has the same label (DataSample.getLabel()) as the given label **int** label.
* If you have *n* samples, e.g. , their sample mean is given by. Note that  is the number of samples, not the number of attributes.
  + **public** **double**[] getStd(**int** label); (Get the standard derivation for class label).

***Hint***: You need to calculate the standard derivation for class label.

* You should loop through all the attributes, and calculate the standard derivation for each attribute.
* You should have an inner loop to loop through all samples. For each sample, you should check whether it has the same label (DataSample.getLabel()) as the given label **int** label.
* If you have *n* samples, e.g. , their standard derivation  is given by, where is the sample mean.
  + **public** DataSample[] getDataSet(); (Get the DataSet dataArray).

1. You should build one class called DataClassification **extends** DataSet that is used to classify test samples.

* Inside this class, you should have the following members:
  + **private** DataSet newData; (represent the array of data samples obtained from “testwine.txt”).
* Inside this class, you should have the following methods:
  + **public** DataClassification(String trainFileName, String newDataFileName); (Constructor for this class.)

***Hint***: It will consist of two statements:

**super**(trainFileName); // **this**.getDataSet() will be the train data // from “wine.txt”

newData = **new** DataSet(newDataFileName); // newData will be the //test data from “testwine.txt”

* + **public** DataSet getNewData(); (Get test data.)
  + **public** **int** nnClassification(DataSample datasp); (classify DataSample datasp using the nearest neighbor classifier.)

***Hint***: It will classify the DataSample datasp by calculating the distance from datasp to all the samples in **this**.getDataSet(), i.e., all data from “wine.txt”, and choose the one that is closest to datasp, and return its label as the label of datasp.

1. You should build one class called main to call **public** **static** **void** main(String[] args).

**Task-related Guidelines and Requirements**

Below is the task-related guideline and requirements.

**Task 1**: Load the data inside the file “wine.txt” into the program.

This task tests your knowledge on file I/O, string parsing and data structure.

The format of the input file is as below:

* It is a text file, lines are separated by a single '\n' character.
* Each sample is one line of text, consisting of 14 entries.
* Those 14 entries are

0) Class label

1) Alcohol

2) Malic acid

3) Ash

4) Alcalinity of ash

5) Magnesium

6) Total phenols

7) Flavanoids

8) Nonflavanoid phenols

9) Proanthocyanins

10)Color intensity

11)Hue

12)OD280/OD315 of diluted wines

13)Proline

The first entry “0) Class label” indicates which type of wine it is. (There are three types of wines, e.g. the class label is one of {1, 2, 3})

* If the format of the file is invalid, the program should catch this exception and print an error message.

You should call DataSet train = **new** DataSet("wine.txt"); to load the data into the program.

**Task 2**: Calculate the basic statistics of the data inside “wine.txt”, and output the statistics into a file named “task2Result.txt”.

This task tests your knowledge on accessing members and methods of a class, data manipulation and write data into a file.

* You should call **this**.getMean(label) and **this**.getStd(label) to get the mean and standard derivation for each class.
* You should then write the string for sample statistics, taskOutputString, into the file named “task2Result.txt”. taskOutputString is defined as:

String taskOutputString =

"Mean of class 1: " + Arrays.*toString*(**this**.getMean(1)) + "\n"

+"Std of class 1: " + Arrays.*toString*(**this**.getStd(1)) + "\n"

+ "Mean of class 2: " + Arrays.*toString*(**this**.getMean(2)) + "\n"

+ "Std of class 2: " + Arrays.*toString*(**this**.getStd(2)) + "\n"

+ "Mean of class 3: " + Arrays.*toString*(**this**.getMean(3)) + "\n"

+ "Std of class 3: " + Arrays.*toString*(**this**.getStd(3)) + "\n";

**Task 3**: Load the test samples from the file named “testwine.txt” into the program.

You should load the test sample in main () as:

DataClassification newtest = **new** DataClassification("wine.txt","testwine.txt");

**Task 4**: Use the nearest-neighbor-classifier to classify the test samples, and output the classification results into a file named “task4Result.txt”.

You should make use of **public** **int** nnClassification(DataSample datasp); to classify the test samples. You need to loop through all test samples inside "testwine.txt".

The output string should be:

String taskOutputString = “The ” + Integer.*toString*(SampleIndex+1) + “-th new sample belongs to class " + Integer.*toString*(testSampleLabel) + ‘\n’; where SampleIndex is the index for test sample (starting from 0, 1, 2, ...), and testSampleLabel is the label of the test sample returned by nnClassification(). You need to loop through all test samples and print all the results into the file named “task4Result.txt”.