# Description of program

This Decision Tree program implementing the Iterative Dichotomiser 3 (id3) algorithm is written in Java using object oriented concepts.

## The reader Class

The reader class reads csv files as input and produces a two dimensional String array as output. It implements classes from the java.io. package to read the csv file line by line and the java.util.StringTokenizer class to break the data elements in the line by the comma delimiter. With a loop, an array is created for each csv line as an array element, each array element is another array with the delimited line items i.e. the rows attributes.

The reader also has a method to convert the 2D String array to a 2D ArrayList.

An ArrayList inherits the List and Collection interfaces and has additional algorithms for tasks such as searching, sorting, counting the frequency of values and copying elements of the array.

## The DataPoint Class

The DataPoint class is used for a creating a datapoint with all its attribute values. i.e. a row in the dataset. The class has one attribute called attributes of ArrayList type that is created by the constructor. The following code returns the value for the first attribute of the datapoint.

DataPoint point = **new** DataPoint();

point.attributes.get(0)

### The id3 Class

* getSubset method

returns 2D ArrayList of a subset of data based on the value of a domain (column). Achieves this by looping through the dataset, creates a DataPoint object for each row, adds the DataPoint’s ArrayList to the subset ArrayList container if the value of the domain equals the parameter.

* getSymbolValue method

goes through the attribute ArrayList of the 2D domain ArrayList for a value of the domain. If the value does not exist it is added otherwise continues such that the domain ArrayList records unique values only.

* getAllValues method

goes through the values of a column in a dataset and appends the unique values in an ArrayList.

* calculateEntropy method

purely calculates entropy for subset received and return the results. It is called to calculate and .

* decomposeNode method

figures the node which gives the best split base upon a measure of the quality of the split. Information gain is calculated in this method. However, data is split into subset by “getSuset” method. After the first split, data is separated into a number of values belonging to parent class. Then, it iterate itself to find the next best attribute. The iteration process continue until entropy is equal to zero or no more data to split.

* alreadyUsedToDecompose method

check if node alrady used or not yet by checking whether tree node of that object has children value assigned and whether decompositionAttribute is equal to the current considering attribute. Othewise, repletes the iteration sending parent object and traverse back.

* createDecisionTree method

has two proposes. Fist, call decomposeNode method to build the tree. Second, evoke printTree to print tree.

* printTree method

is a method to use to print out the output kept in objects of decision tree program.