Your code is to be accompanied by a short report of maximum 4 pages (plus screenshots) outlining your design and any ways in which your solution goes beyond the original project brief.

Original project brief:

The program will contain the following components:

* **code to read in the data from a file and place it in classes**
  + Processing provides both loadBytes and loadStrings commands. The split method should also be helpful.
  + A simple (although not particularly efficient) solution would be to define a DataPoint class which represents a single space object. There would be one instance of the class for each entry in the input file.
  + You may wish to store the data in a format which is more efficient to access.
* **code to select a subset of this data.**
  + Not all the data will be shown on the screen at one time, and so a set of queries must be defined in your code. At a minimum, the following queries should be implemented:

▪ Objects associated with a particular country (state).

▪ Objects within a certain altitude (perigee/apogee) range

▪ Objects sorted by launch date.

* **code to draw the data to the screen.**
  + The results of each query will need to drawn on the screen.
  + You are encouraged to use graphical representations where appropriate (eg. the data could be on a barchart, a 2D or 3D representation of orbits, or more complex visualisations such as heatmaps).
* **code to handle user commands.**
  + Selecting what data is to be displayed (the query), the country name, mass, launch date range, etc.
* **code to put everything together.**
  + You are advised to have an outline of this as early as possible (first week of project).

Week 1:

This week’s goal

* IMPORTANT: have everyone in the group add their names to a README file using SVN - add(first person) then update/commit (everyone else).
* Download test data from module website.
* Call a method in setup which reads in the data.
* Initially just use println() to verify each line is being read in properly.
* Create a class to store the datapoints – each instance storing the data from one line of the file.
* Create a data structure (e.g. ArrayList) which stores multiple instances of this class.
* Create a loop which prints out all of the instances using println()
* Print out the instances on the screen in a nice font using text().

Week 2:

* sketch out some designs on paper with your group and bring to the lab to discuss with your demonstrator.
* read in the tsv file and store the data (in setup), as for last week.
* create a (simple) query result and set this to be the current query (in setup) write a method which will draw this query result
* as a chart (e.g. bar chart).
* draw the results of this query as a bar chart (in draw), using the method you have defined.

Week 3:

* Implement at least three different queries on the data.
* Have some selection mechanism to invoke the different queries (this can be very basic – eg. press a key).
* Draw the results of the queries on the
* screen.

Week 4:t

…