# Assignment 1 Samuel Clement 12287539

<https://github.com/BustahBoi/BCPR301---Assignment-1>

## Validate Data

I wrote the base of the validator and its methods, including the original Regular Expression statements.

I believe this component is robust as it is able to check each data point individually and then determines if data is valid by checking all 7 check statements return true.   
This component is clever as it can handle the different lengths of months in each year.

## File Handler

TXT Reader method for filehandler class.

This filehandler could be made more robust. Currently it relies on the validator for the keys and due to the validator not handling different input as well as it should then it is likely to cause an issue that does not originate with the filehandler as the file handlers sole purpose is to extract information not to validate such information.

This method is clever as it splits up the line on each txt file by looking for the sets of data. This is done by splitting with the delimiter character (in this case I have used “ ; “ and have left the ability to set that character manually for later revision of the system). I believe this is clever as it does not try and set a limit to the length of the row that can be taken from a file. This means that if an extra field is entered it will not break the system at the file handling level. (this does not mean it will not produce issues if you give it 8 fields to read, it means that problems are created elsewhere rather than at the file handling level)

## Charts / Graphs

I wrote the Graph Abstract class and the 3 concrete graph classes (Scatter, Pie, Bar)

Used Abstract Base Class to avoid duplicating attributes of the concrete graph classes. Each concrete class is just a single method that overwrites the “draw\_graph()” method of the base class.

## Database Handler

Wrote initial database handler class which was later modified by Wesley to add functionality.

This class is robust as it does not allow database objects (remote and local databases) to exist without being a part of a DatabaseHandler. This means that if the handler is deleted the database references are also lost. This means that should a database handling object get deleted the Remote/Local database objects will also be lost. This is an example of dependency so that Local and Remote database objects cannot exist freely in the system without the existence of a handler to contain them.

## Exception & Error Handling

I implemented exception handling on the do\_load method of the prompt file. This looked for exceptions that were produced by invalid file name and path. It also had broad exception handling to catch any uncaught specific exceptions. It also can look for value errors and attribute errors.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Feature | Used by Peers | Robustness | Well Implemented | Marks |
| 2. | Has a CMD | Used by both (2) | (1) | (1) | 4 |
| 3. | Display Command Help | Used by both (2) | (1) | (1) | 4 |
| 5. | Validate Data | Used by both (2) | (1) | (1) | 4 |
| 7. | Can Load from file | Used by both (2) | (1) | (1) | 4 |
| 8. | Exceptions | Used by none (0) | (1) | (1) | 2 |
| 9. | Error trapping | Used by none (0) | (1) | (1) | 2 |
| 10 | Provide Doctests | Used by none (0) | (1) | (1) | 2 |
| 11. | Provide Unittests | Used by none (0) | (1) | (1) |  |
| 13. | Directories & Locations | Used by both (2) | (1) | (1) | 4 |
| 14. | Charts/Graphs | Used by both (2) | (1) | (1) | 4 |
| 15. | Database (handler) | Used by both (2) | (1) | (1) | 4 |