REPORT

Data Validation approach:
The only validations needed were for characters, integers and doubles. No validations were done for strings as string itself is a combination of numbers and letters. Hence, even "As123You" will be considered as string. The reason why I created a separate static class for the validation methods is to avoid placing all the validation submodules into every class created.

Justification for Design Decisions:
AssignmentTestHarness :-
This is the main class that will be run. A main menu is provided here to show readability and options that will be executed when chosen to understand the main purpose of the program without having to go through each and every class created.

STATIC CLASSES:
MenuClass :-
This static class provides further sub-menus for the user. For example: "(P)NG or (C)SV?". The reason I decided to create a separate sub-menu is due to the fact that there are a few main menu options that require to display the same options to the user. One perfect example would be the options -- exporting to file and importing an image-- in which both will require the user to choose a PNG or CSV file for import or export.

UserInterface :-
The main purpose of this class is to handle the prompts that is to be displayed to the user. Input and output operations (such as entering an integer, string or even a real number) will have to go through this class before the value is returned to the main. Operations such as having to display an array or to display any errors encountered will be handled by this class

DetectEdges :-
I have created a separated class specifically for performing the convolution and smoothing operation of an image. This is purely for readability in the main. Inside this class contains submodules to help with the calculation of the smoothing and convolution formulas.

FileIO :-
This class handles any file operations including reading and writing both images of PNG type and files of CSV type. Having a separate class for handling files are much more efficient as files have their own Scanner class which in their class is called a printwriter. This class also contains a submodule to convert texts in CSV file to a readable and suitable format for image or 2D arrays and vice versa.

PDIMath :-
This class is used to perform any mathematic calculations required by the program. Some of these includes using the ceil function, floor, abs and many more. I have decided to place the mathematical calculations in a separate static class to avoid any messy codes in the main or even in the other respective classes.

Validation :-
I have created a separate class for validation for the main reason that I have stated above.

MODEL CLASSES:
Image :-
This class is created for obtaining simplicity in the main. The reason why I have decided to create a model class for image is because there can be several image that may require the same usage of functions. If this model class had not been created, I will have to declare multiple 2D Integer array which in this case will be inefficient. With this model class, I can just create several instances of type Image.

Date :-
I have decided to create a separate model class for Date with the main intent to allow different dates to be created and still have the same day, month and year validation under the same class. This also makes it easier for the string date to be output to the user in a readable format for example, 2020-04-05 in which 2020 is the year, 4 is the month and 5 is the day.

Explanation for the usage of static classes and model objects :-
Static classes are advantageous as it does not need the creation of objects just to call on its methods. However, it does have restrictions such as not being able to invoke methods that are non-static. If I just want to find the lesser value of two integers, a static class would be better as I do not need to keep creating a new instance of the class just to store a result.

The reason why model classes are used is to create a main blueprint for objects that will require to use the same methods in the class itself. One obvious advantage of using model classes is the ability for objects to have the same attributes. If I am only using model classes for this assignment, it will require a much larger amount of memory compared to if I have separated some classes to be an instance class and the rest to be static classes.

Real-world application :-
Some real world of edge detection can be for face recognition as security measures, finger-print recognition and even used in medical science. By using edge detection, unknown or foreign objects such as tumors can be located and detected in an organ of the patient.

Challenges in implementation :-
The biggest challenge I had took me three day. Initially, my edge detection algorithm could only work properly with the Vertical Kernel and any other kernels would have not been successful to detect the edges of my image. I decided to check on my readFile method (currently called as csvFile for specificity) and debugged the code line by line. I found that as the kernel was converted to a 2D array of integers, the values in the array seemed to be corrupted. Apparently, I had designed the algorithm in a way where it is biased towards the final row values in the file. Meaning to say, that every value in the array would only store in the final row values in the kernel file. And that was why, my code could not work perfectly. To clarify further, the reason why it had worked with the vertical kernel was because every row had the same values, so detecting the error was not easy. For example, my first row have values 1,0,-1 and my second and third row would also have values in the same order (1,0,-1).