2. (15 points) **Section B (Scratch Implementation)**

    1. (2 marks) Create a dataset (with 10,000 points) using the circle equation

       $$(x - h)^2 + (y - k)^2 = r^2$$

       such that: h=0, k=0, r=1, with the label **0** and h=0, k=3, r=1, with the label **1**.

       - Write a class named dataset which takes number of points as input.
       - The class should have a function named get(add_noise=False), which should give a set of pre-defined number of points. Every call to this function returns random points, given it satisfies the conditions above. DO NOT implement this class in the main .py/.ipynb file instead create a separate utils.py and import the functions you need in main file.
       - Given, add_noise=True, as an input to the get function, it should also add Gaussian noise with mean 0 and standard deviation 0.1.

    2. (1 mark) Plot the dataset on a 2-D plot such that all the information related to the dataset i.e., x, y, and labels can be inferred from the 2d plot itself with add_noise argument set to True and False.

    3. (5 marks) Train a classifier using the Perceptron training algorithm (PTA), taught in class, on the data you just created (with and without noise) and plot the decision boundary if there exists one. Otherwise explain why not a decision boundary exists. NOTE: You have implement the PTA algorithm as a python class yourself from scratch using only numpy and python. DO NOT implement this class in the main .py/.ipynb file instead implement this in the utils.py and import the class/functions you need in main file.

    4. (3 marks) Train another classifier using the perceptron training algorithm (PTA) on the data you just created (without noise) but with a fixed bias equal to "0" and plot the decision boundary if there exists one. Compare the results with question 2.3 and write a brief report of at least 150 words.

    5. (3 marks) Create a dataset (with 4 points) using the XOR, AND, and OR property. Plot decision boundary, if there exists one, using the PTA such that the bias is learnable, and fixed (equals to "0")