TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

Central Department of Computer Science and Information Technology

Kirtipur, Kathmandu

Lab No.: 1

A Lab Report on *Point Line Classification*

Submitted by:

Name: Brihat Ratna Bajracharya
Roll No.: 19/075

Submitted to:

Mr. Jagdish Bhatta
Central Department of Computer Science
and Information Technology

Date of submission: *2076 Kartik 28*

LAB 1

Point Line Classification (PLC)

1. Implement Point
2. Implement Line Segment
3. Test for 5 dimensions of PLC

Code (https://github.com/Brihat9/CG/blob/master/cg_lab_1_plc.py)

```
#!/usr/bin/env python

class Point:
    """ Point Class """

    def __init__(self, xcoord, ycoord):
        self.x = xcoord
        self.y = ycoord

    @classmethod
    def input_point(point):
        """ Takes X-Coord and Y-Coord from user to form a point """
        return point(
            int(raw_input('  X-Coord: ')),
            int(raw_input('  Y-Coord: ')),
        )

    def __str__(self):
        """ Displays point's coordinates """
        return "(" + str(self.x) + ", " + str(self.y) + ")"

    def __eq__(self, other):
        """ Compares two instances of this Class """
        if not isinstance(other, Point):
            # comparision against unrelated types
            return NotImplemented

        return self.x == other.x and self.y == other.y

class LineSegment:
    """ Line Segment Class """
    def __init__(self, point1, point2):
        self.start = point1
        self.terminal = point2

    def __str__(self):
        """ Displays end-points of line segment """
        return "[" + str(self.start) + ", " + str(self.terminal) + "]"

def check_colinear(point1, point2, point3):
    """ Checks whether three points are colinear or not
        point1(A) and point2(B) are endpoints, check point3(P) for colinearity
        USING slope based approach, check slope of line AB == slope of line AP
    """
    if((point2.y - point1.y) * (point3.x - point1.x) == (point3.y - point1.y) * (point2.x -
point1.x)):
        return True
    return False

def check_plc(start_point, terminal_point, qpoint):
    """ Checks PLC for given given query point and line segment """
```

```python
    if qpoint.x == start_point.x and qpoint.y == start_point.y:
        print(" Query Point " + str(qpoint) + " is Start Point")
    elif qpoint.x == terminal_point.x and qpoint.y == terminal_point.y:
        print(" Query Point " + str(qpoint) + " is Terminal Point")
    elif check_colinear(start_point, terminal_point, qpoint):
        line_segment = LineSegment(start_point,terminal_point)
        print("\nFor line segment with end-points " + str(line_segment))

        """ case for vertical line segment (parallel to Y-Axis) """
        if start_point.x == terminal_point.x and start_point.y != terminal_point.y:
            print(" Note: Line segment parallel to Y-axis")

            if qpoint.y > terminal_point.y:
                print(" Query Point " + str(qpoint) + " is Beyond Line Segment")
            elif qpoint.y < start_point.y:
                print(" Query Point " + str(qpoint) + " is Behind Line Segment")
            elif qpoint.y > start_point.y and qpoint.y < terminal_point.y:
                print(" Query Point " + str(qpoint) + " is between Start and Terminal Point")
        else:
            if qpoint.x > terminal_point.x:
                print(" Query Point " + str(qpoint) + " is Beyond Line Segment")
            elif qpoint.x < start_point.x:
                print(" Query Point " + str(qpoint) + " is Behind Line Segment")
            elif qpoint.x > start_point.x and qpoint.x < terminal_point.x:
                print(" Query Point " + str(qpoint) + " is between Start and Terminal Point")
            else:
                print(" This case not considered")
    else:
        print("\n Query Point " + str(qpoint) + " not colinear with line segment")

def main():
    """ Main Function """

    print("CG LAB 1 (Point Line Classification v2_colinearity_check)")
    print("Brihat Ratna Bajracharya\n19/075\n")

    print("Enter end-points of Line Segment")
    print(" Start Point")
    start_point = Point.input_point()
    print("\n Terminal Point")
    terminal_point = Point.input_point()

    if start_point == terminal_point:
        print("\n Start Point and Terminal Point found same.")
    else:
        ''' swap start and terminal point if x-coord of start point is
            greater than x-coord of terminal point '''
        if start_point.x > terminal_point.x:
            start_point, terminal_point = terminal_point, start_point

        print("\nEnter Query Point")
        query_point = Point.input_point()

        check_plc(start_point, terminal_point, query_point)

    print("\nDONE.\n")

    ''' TEST CASES '''
    """ test for colinearity function """
    # if(check_colinear(start_point, terminal_point, query_point)):
    #     print("3 points colinear")
    # else:
    #     print("3 points not colinear")

    """ test for 5 dimensions of PLC """
    # qbehind = Point(1,2)
    # qbeyond = Point(8,4)
    # qbetween = Point(5,6)
    # qstart = Point(2,3)
    # qterminal = Point(7,9)
    #
    # testcases = [qbehind, qbetween, qbeyond, qstart, qterminal]
    #
    # for qpoint in testcases:
    #     check_plc(start_point, terminal_point, qpoint)

if __name__ == '__main__':
    main()
```

Output 1:

```
$ ./cg_lab_1_plc.py
CG LAB 1 (Point Line Classification v2_colinearity_check)
Brihat Ratna Bajracharya
19/075

Enter end-points of Line Segment
 Start Point
  X-Coord: 2
  Y-Coord: 4

 Terminal Point
  X-Coord: 8
  Y-Coord: 16

Enter Query Point
  X-Coord: 3
  Y-Coord: 6

For line segment with end-points [(2, 4), (8, 16)]
 Query Point (3, 6) is between Start and Terminal Point

DONE.
```

Output 2:

```
$ ./cg_lab_1_plc.py
CG LAB 1 (Point Line Classification v2_colinearity_check)
Brihat Ratna Bajracharya
19/075

Enter end-points of Line Segment
 Start Point
  X-Coord: 3
  Y-Coord: 6

 Terminal Point
  X-Coord: 6
  Y-Coord: 12

Enter Query Point
  X-Coord: 2
  Y-Coord: 4

For line segment with end-points [(3, 6), (6, 12)]
 Query Point (2, 4) is Behind Line Segment

DONE.
```