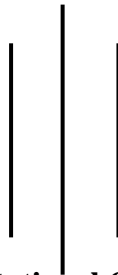


**Tribhuvan University
Institute of Science and Technology**



**Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu**



**Computational Geometry
Assignment**

Lab 1: Implementation of Geometric Objects

Submitted by:

Name: Raju Shrestha

Roll No.: 48/2079

Date: 01-October-2023

Submitted To:

Asst. Prof. Jagdish Bhatta

CDCSIT

Lab 1: Write a program to implement the following geometric objects.

1. Point
2. Line Segment
3. Ray
4. Line

1. Program to Implement Point

```
import matplotlib.pyplot as plt

class Node:
    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def add_point(self, x, y):
        new_point = Node(x, y)
        new_point.next = self.head
        self.head = new_point
    def display_points(self):
        current = self.head
        while current:
            print(f"({current.x}, {current.y})")
            current = current.next
    def get_points(self):
        points = []
        current = self.head
        while current:
            points.append((current.x, current.y))
            current = current.next
        return points

list = LinkedList()
num_points = int(input("Number of points: "))
for _ in range(num_points):
    x = float(input("Enter x coordinate: "))
    y = float(input("Enter y coordinate: "))
    list.add_point(x, y)

list.display_points()

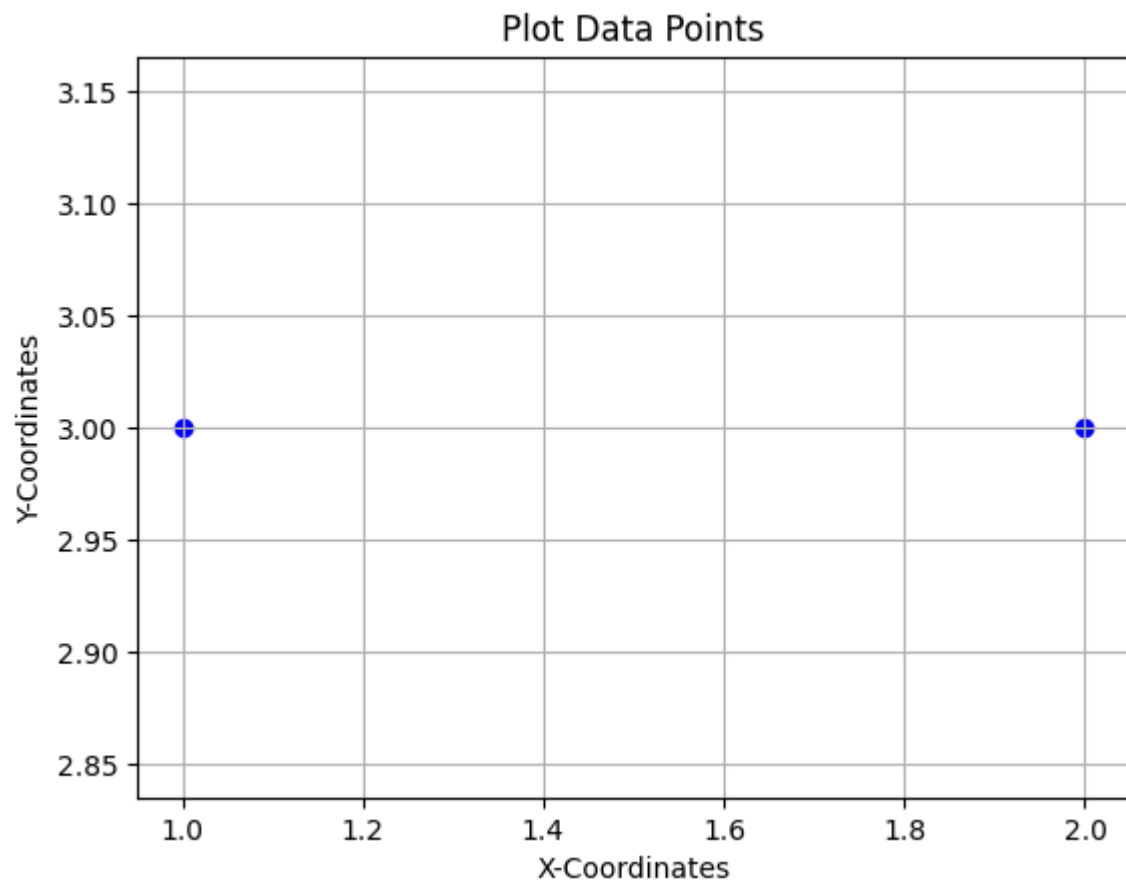
points = list.get_points()
x, y = zip(*points)

plt.scatter(x, y, color='blue')
plt.title('Plot Data Points')
```

```
plt.xlabel('X-Coordinates')  
plt.ylabel('Y-Coordinates')  
plt.grid(True)
```

Output:

```
Number of points: 2  
Enter x coordinate: 2  
Enter y coordinate: 3  
Enter x coordinate: 1  
Enter y coordinate: 3  
(1.0, 3.0)  
(2.0, 3.0)
```



2. Program to Implement Line Segment

```
import matplotlib.pyplot as plt
class Node:
    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def add_point(self, x, y):
        new_point = Node(x, y)
        new_point.next = self.head
        self.head = new_point
    def display_points(self):
        current = self.head
        while current:
            print(f"({current.x}, {current.y})")
            current = current.next
    def get_points(self):
        points = []
        current = self.head
        while current:
            points.append((current.x, current.y))
            current = current.next
        return points

list = LinkedList()
for _ in range(int(input("Number of points: "))):
    list.add_point(float(input("Enter x-coordinate: ")),
float(input("Enter y-coordinate: ")))

list.display_points()

points = list.get_points()
x, y = zip(*points)

plt.scatter(x, y, color='blue')

plt.plot(x, y, linestyle='-', color='red', marker='o')

plt.title('Plot Data Points with Line Segments')
plt.xlabel('X-Coordinates')
plt.ylabel('Y-Coordinates')
plt.grid(True)
plt.show()
```

Output:

Number of points: 2

Enter x-coordinate: 2

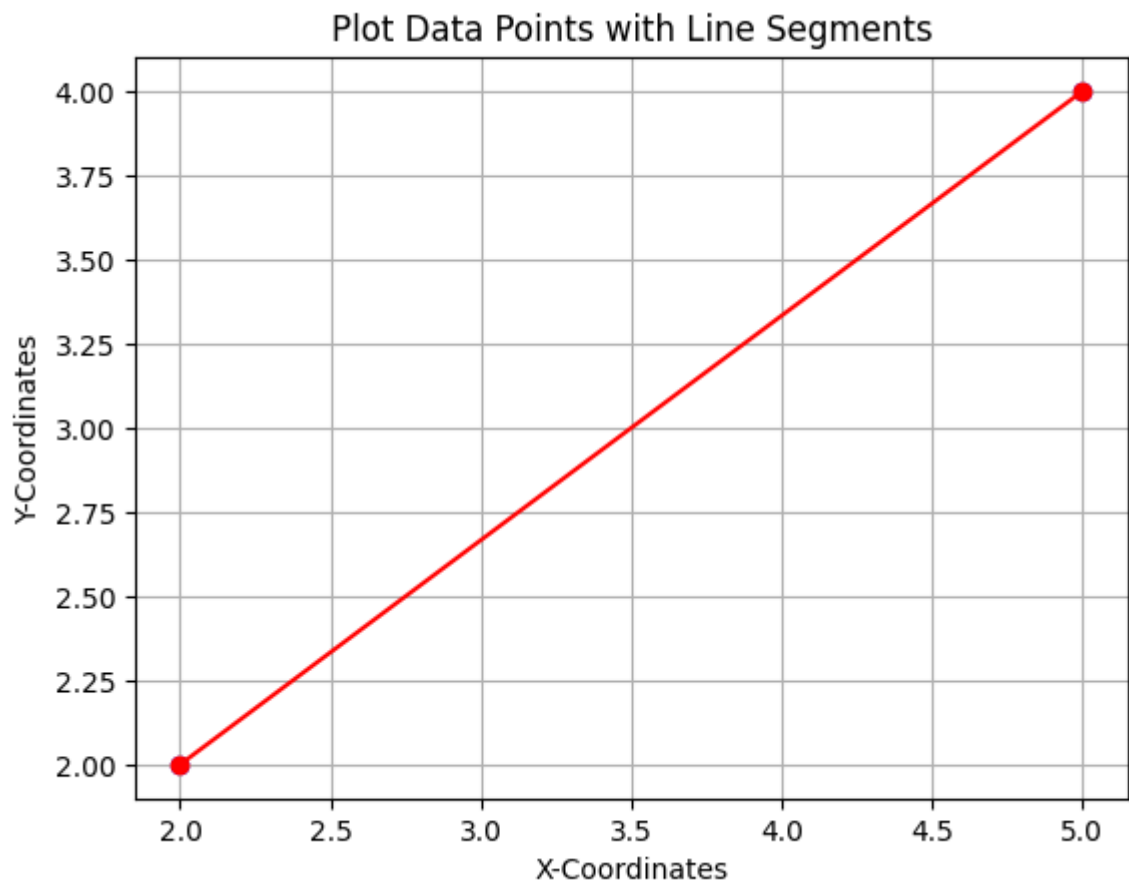
Enter y-coordinate: 2

Enter x-coordinate: 5

Enter y-coordinate: 4

(5.0, 4.0)

(2.0, 2.0)



3. Program to Implement Ray

```
import matplotlib.pyplot as plt
class Node:
    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def add_point(self, x, y):
        new_point = Node(x, y)
        new_point.next = self.head
        self.head = new_point
    def display_points(self):
        current = self.head
        while current:
            print(f"({current.x}, {current.y})")
            current = current.next
    def get_points(self):
        points = []
        current = self.head
        while current:
            points.append((current.x, current.y))
            current = current.next
        return points

list = LinkedList()
for _ in range(int(input("Number of points: "))):
    list.add_point(float(input("Enter x-coordinate: ")),
float(input("Enter y-coordinate: ")))

list.display_points()
points = list.get_points()
x, y = zip(*points)
plt.scatter(x, y, color='blue')
common_origin_x, common_origin_y = x[0], y[0]
for i in range(1, len(x)):
    plt.quiver(common_origin_x, common_origin_y, x[i] -
common_origin_x, y[i] - common_origin_y,
angles='xy', scale_units='xy', scale=1, color='red',
width=0.005)
plt.title('Plot Data Points with Rays')
plt.xlabel('X-Coordinates')
plt.ylabel('Y-Coordinates')
plt.grid(True)
plt.show()
```

Output:

Number of points: 2

Enter x-coordinate: 4

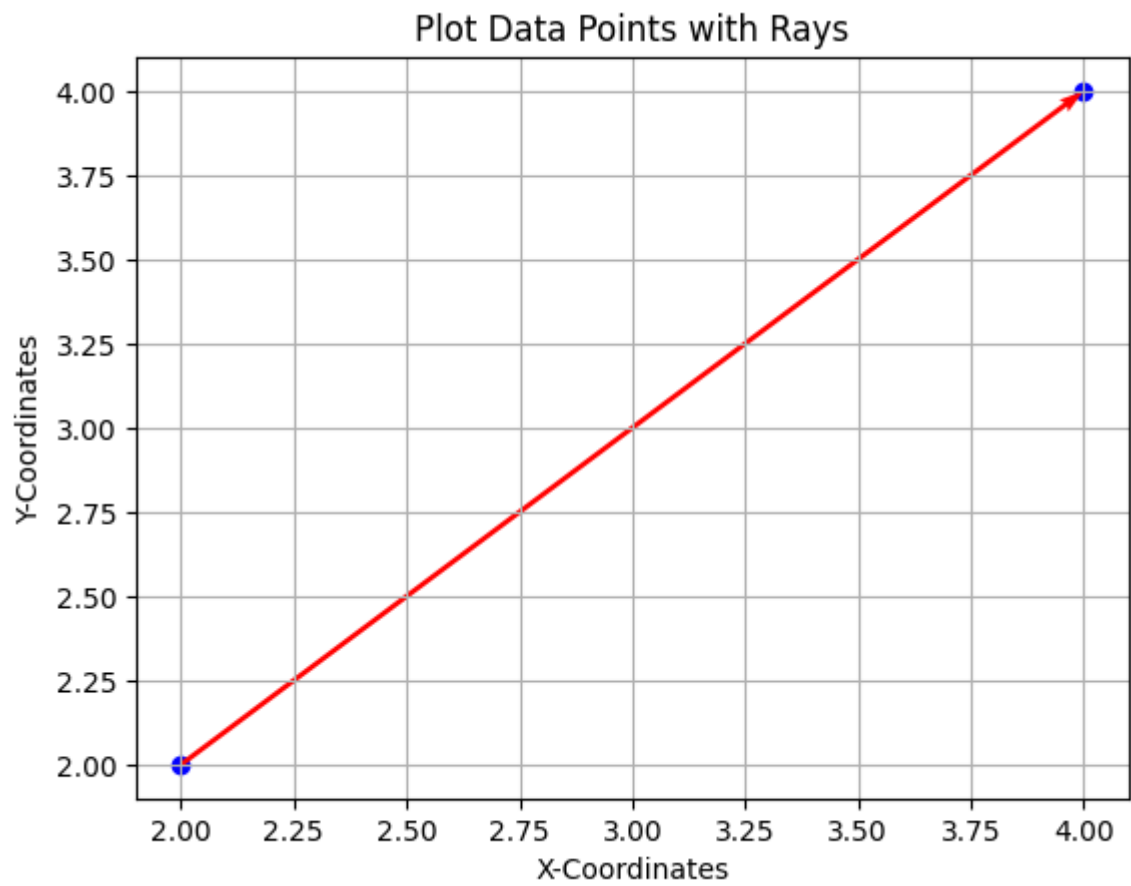
Enter y-coordinate: 4

Enter x-coordinate: 2

Enter y-coordinate: 2

(2.0, 2.0)

(4.0, 4.0)



4. Program to Implement Line

```
import matplotlib.pyplot as plt
class Node:
    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def add_point(self, x, y):
        new_point = Node(x, y)
        new_point.next = self.head
        self.head = new_point
    def display_points(self):
        current = self.head
        while current:
            print(f"({current.x}, {current.y})")
            current = current.next
    def get_points(self):
        points = []
        current = self.head
        while current:
            points.append((current.x, current.y))
            current = current.next
        return points
list = LinkedList()
num_points = int(input("Number of points: "))
for i in range(1, num_points + 1):
    x_coord = float(input(f"Enter x-coordinate for P{i}: "))
    y_coord = float(input(f"Enter y-coordinate for P{i}: "))
    list.add_point(x_coord, y_coord)
list.display_points()
points = list.get_points()
x, y = zip(*points)
center_x, center_y = sum(x) / len(x), sum(y) / len(y)
plt.scatter(x, y, color='blue')
for i in range(len(x)):
    end_x = center_x + (center_x - x[i]) * 5
    end_y = center_y + (center_y - y[i]) * 5
    plt.plot([center_x, end_x], [center_y, end_y], linestyle='--',
color='green')
    plt.annotate(f'P{i}', (x[i], y[i]), textcoords="offset points",
xytext=(0, 5), ha='center', fontsize=8)
plt.title('Plot Data Points with Lines')
plt.xlabel('X-Coordinates')
plt.ylabel('Y-Coordinates')
plt.grid(True)
```



```
plt.show()
```

Output:

Number of points: 2

Enter x-coordinate for P1: 2

Enter y-coordinate for P1: 2

Enter x-coordinate for P2: 6

Enter y-coordinate for P2: 6

(6.0, 6.0)

(2.0, 2.0)

