Point Cloud Learning Assignment (Day 13 -15)

Q1. Segment and Color Point Cloud Using DBSCAN

- Load the provided .pcd file using Open3D
- Apply DBSCAN clustering:
  Use : labels = np.array(pcd.cluster_dbscan(eps=0.1, min_points=5))
- Assign a different color to each cluster
- Mark noise points (label = -1) in black
- Visualize using draw_geometries()

Submit:

- Screenshot showing the segmented and colored point cloud
- Mention the eps and min_points used

Q2. Explain Segmentation and Clustering

In 2–3 lines each, answer:

- What is segmentation in point cloud processing?
- How does clustering help in segmentation?
- Mention one difference between DBSCAN and Euclidean clustering
- Provide one real-world example of where segmentation is useful

Q3. Visualize Colored Clusters

- After clustering, use a colormap (e.g., tab20) to color the clusters
- Use the matplotlib colormap to assign colors:
  colors = plt.get_cmap("tab20")(labels / (max_label + 1))[:, :3]
- Apply colors to pcd.colors and visualize

Submit:

- Screenshot showing clearly colored clusters
- Mention the colormap used (e.g., tab20)

Q4. Add a Bounding Box as Annotation

- Choose any one cluster
- Extract it using select_by_index()
- Add a bounding box annotation (either AABB or OBB):
  Use: box = cluster_pcd.get_axis_aligned_bounding_box()
  box.color = (1, 0, 0) # Red box
- Display the cluster with the box

Submit:

- Screenshot showing the bounding box on the cluster
- Write a short caption: "This box shows a visual boundary of a detected cluster."

Q5. Add a Line Annotation Between Two Points

- Select any two 3D points from your point cloud (manually or by index)
- Use LineSet to draw a line between them:

```
line = o3d.geometry.LineSet(
    points=o3d.utility.Vector3dVector([point1, point2]),
    lines=o3d.utility.Vector2iVector([[0, 1]])
)
```

Submit:

- Screenshot showing the line annotation
- Mention the 3D coordinates or indices of the two points