**Section 9**: Finding a Central Location  
Overview  
Have you ever wondered where the most centrally located point on campus is? How about a centrally located point between you and a  
number of your friends? In this Problem Set, you’ll write a JavaFX program that answers these questions visually.  
Tasks  
Review CampusMap.mp4 from Section 9, Lesson 1, Slide 6.  
Your goal is to create the CampusMap program that uses your map of campus,  
dorm names, dorm populations, and your group of friends. You’re welcome to  
design your own campus map (this is your background graphic). You’ll have to  
design your own campus map if your actual campus has fewer than 3 dorms,  
otherwise this Problem Set wouldn’t be too interesting.  
The Dorms  
Choose a way to visually represent dorms. The name and population of a dorm  
must also be visible. The population and location of each dorm must somehow be  
adjustable while the program is running.  
The Center Points  
Your program must show two center points. The first point represents the central location of all students in all dorms. This is  
essentially a center of mass problem where dorms with a larger population are considered more “massive” and have a greater influence  
over the center point’s location.  
The second point represents the central location of your study group. Create a study group of at least 3 people, 1 of which must live in  
a different dorm.  
Both center points must include a visual representation, a label, and display their location as numeric values. These points should  
automatically update as a dorm’s location or population changes. You’re welcome to leave these measurements as pixels or convert  
them into real-life units of distance.  
However you choose to represent your dorms and points, remember to perform your distance calculations based on the geometric  
center of these visuals, and not the top-left corners.  
Hints :  
There are certain concepts we didn’t cover in Section 9, like how to work with a Text node. But we did discuss how to consult the  
JavaFX API Documentation. Part of the challenge of this Problem Set is understanding how to consult resources. If you have ideas  
about a feature you’d like to implement or a technique you’d like to explore, don’t be afraid to consult the JavaFX API Documentation. It  
has a lot of fun things to show you.

**PROGRAM :**

public class Dorm {

private String name;

private int population;

private double x, y; // Coordinates on the map

public Dorm(String name, int population, double x, double y) {

this.name = name;

this.population = population;

this.x = x;

this.y = y;

}

// Getters and Setters

public String getName() {

return name;

}

public int getPopulation() {

return population;

}

public void setPopulation(int population) {

this.population = population;

}

public double getX() {

return x;

}

public void setX(double x) {

this.x = x;

}

public double getY() {

return y;

}

public void setY(double y) {

this.y = y;

}

}

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.AnchorPane;

import javafx.scene.paint.Color;

import javafx.scene.shape.Circle;

import javafx.stage.Stage;

import java.util.ArrayList;

public class CampusMap extends Application {

private ArrayList<Dorm> dorms = new ArrayList<>();

private Circle campusCenter;

private Circle studyGroupCenter;

private Label campusCenterLabel;

private Label studyGroupCenterLabel;

@Override

public void start(Stage primaryStage) {

// Setup UI components

AnchorPane root = new AnchorPane();

// Background Image

ImageView campusMap = new ImageView(new Image("campus\_map.png")); // Replace with your campus map image

root.getChildren().add(campusMap);

// Dorms initialization (Example)

Dorm dorm1 = new Dorm("Dorm 1", 100, 200, 150);

Dorm dorm2 = new Dorm("Dorm 2", 120, 400, 300);

Dorm dorm3 = new Dorm("Dorm 3", 90, 300, 500);

dorms.add(dorm1);

dorms.add(dorm2);

dorms.add(dorm3);

// Create circles and labels for dorms

for (Dorm dorm : dorms) {

Circle dormCircle = new Circle(dorm.getX(), dorm.getY(), 10, Color.BLUE);

Label dormLabel = new Label(dorm.getName() + ": " + dorm.getPopulation());

dormLabel.setLayoutX(dorm.getX());

dormLabel.setLayoutY(dorm.getY() - 20);

root.getChildren().addAll(dormCircle, dormLabel);

}

// Campus Center Circle

campusCenter = new Circle(0, 0, 10, Color.RED);

campusCenterLabel = new Label("Campus Center: (0, 0)");

root.getChildren().addAll(campusCenter, campusCenterLabel);

// Study Group Center Circle

studyGroupCenter = new Circle(0, 0, 10, Color.GREEN);

studyGroupCenterLabel = new Label("Study Group Center: (0, 0)");

root.getChildren().addAll(studyGroupCenter, studyGroupCenterLabel);

// Calculate and update center points

updateCenterPoints();

Scene scene = new Scene(root, 800, 600);

primaryStage.setScene(scene);

primaryStage.setTitle("Campus Map");

primaryStage.show();

}

private void updateCenterPoints() {

double totalX = 0, totalY = 0;

int totalPopulation = 0;

for (Dorm dorm : dorms) {

totalX += dorm.getX() \* dorm.getPopulation();

totalY += dorm.getY() \* dorm.getPopulation();

totalPopulation += dorm.getPopulation();

}

double campusCenterX = totalX / totalPopulation;

double campusCenterY = totalY / totalPopulation;

campusCenter.setCenterX(campusCenterX);

campusCenter.setCenterY(campusCenterY);

campusCenterLabel.setLayoutX(campusCenterX);

campusCenterLabel.setLayoutY(campusCenterY + 20);

campusCenterLabel.setText("Campus Center: (" + (int)campusCenterX + ", " + (int)campusCenterY + ")");

// Assuming the study group center is calculated similarly (for now, using all dorms)

studyGroupCenter.setCenterX(campusCenterX + 50); // Offset for demo purposes

studyGroupCenter.setCenterY(campusCenterY + 50);

studyGroupCenterLabel.setLayoutX(campusCenterX + 50);

studyGroupCenterLabel.setLayoutY(campusCenterY + 70);

studyGroupCenterLabel.setText("Study Group Center: (" + (int)studyGroupCenter.getCenterX() + ", " + (int)studyGroupCenter.getCenterY() + ")");

}

public static void main(String[] args) {

launch(args);

}

}