CAMERA RENTAL APPLICATION PROGRAM:

```
import java.util.*;

class Camera {
    private static int idCounter = 1;

    private int id;
    private String brand;
    private String model;
    private double perDayPrice;

    public Camera(String brand, String model, double perDayPrice) {
        this.id = idCounter++;
        this.brand = brand;
        this.model = model;
        this.perDayPrice = perDayPrice;
    }

    public int getId() {
        return id;
    }

    public String getBrand() {
        return brand;
    }

    public String getModel() {
        return model;
    }
```

```
    public double getPerDayPrice() {
        return perDayPrice;
    }
}

class CameraRental {
    private List<Camera> cameraList;
    private double walletBalance;

    public CameraRental() {
        cameraList = new ArrayList<>();
        walletBalance = 0.0;
    }

    public void addCamera(Camera camera) {
        cameraList.add(camera);
        System.out.println("Your camera has been successfully added to the list.");
    }

    public void removeCamera(int cameraId) {
        boolean removed = false;
        Iterator<Camera> iterator = cameraList.iterator();
        while (iterator.hasNext()) {
            Camera camera = iterator.next();
            if (camera.getId() == cameraId) {
                iterator.remove();
                removed = true;
                break;
            }
        }
```

```
    if (removed) {
        System.out.println("Camera successfully removed from the list.");
    } else {
        System.out.println("Camera with the specified ID not found.");
    }
}

public void viewMyCameras() {
    if (cameraList.isEmpty()) {
        System.out.println("No cameras are currently added to the list.");
    } else {
        System.out.println("My Cameras:");
        for (Camera camera : cameraList) {
            System.out.println("ID: " + camera.getId() +
                               ", Brand: " + camera.getBrand() +
                               ", Model: " + camera.getModel() +
                               ", Per Day Price: " + camera.getPerDayPrice() + " INR");
        }
    }
}

public void rentCamera(int cameraId) {
    Camera selectedCamera = null;
    for (Camera camera : cameraList) {
        if (camera.getId() == cameraId) {
            selectedCamera = camera;
            break;
        }
    }
    if (selectedCamera == null) {
        System.out.println("Camera with the specified ID not found.");
```

```
    } else {
        if (walletBalance >= selectedCamera.getPerDayPrice()) {
            walletBalance -= selectedCamera.getPerDayPrice();
            System.out.println("Your transaction for the camera has been successfully completed.");
        } else {
            System.out.println("Insufficient balance in your wallet to rent this camera.");
        }
    }
}

public void viewAllCameras() {
    if (cameraList.isEmpty()) {
        System.out.println("No cameras are currently available for rent.");
    } else {
        System.out.println("Available Cameras:");
        for (Camera camera : cameraList) {
            System.out.println("ID: " + camera.getId() +
                               ", Brand: " + camera.getBrand() +
                               ", Model: " + camera.getModel() +
                               ", Per Day Price: " + camera.getPerDayPrice() + " INR");
        }
    }
}

public void viewWalletBalance() {
    System.out.println("Your current wallet balance is " + walletBalance + " INR.");
}

public void depositToWallet(double amount) {
    walletBalance += amount;
    System.out.println("Your wallet balance has been updated successfully. Current balance is " +
```

```
                walletBalance + " INR.");
    }

    public void run() {
        Scanner scanner = new Scanner(System.in);
        int choice;
        do {
            System.out.println("\nWelcome to the Camera Rental App");
            System.out.println("Please login to continue");
            System.out.print("Username: ");
            String username = scanner.nextLine();

            System.out.println("\nMain Menu:");
            System.out.println("1. My Cameras");
            System.out.println("2. Rent a Camera");
            System.out.println("3. View All Cameras");
            System.out.println("4. My Wallet");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character

            switch (choice) {
                case 1:
                    System.out.println("\nMy Cameras Menu:");
                    System.out.println("a. Add");
                    System.out.println("b. Remove");
                    System.out.println("c. View My Cameras");
                    System.out.println("d. Go to Previous Menu");
                    System.out.print("Enter your choice: ");
                    char subChoice = scanner.nextLine().charAt(0);
```

```
    switch (subChoice) {
        case 'a':
            System.out.print("Enter the camera brand: ");
            String brand = scanner.nextLine();
            System.out.print("Enter the model: ");
            String model = scanner.nextLine();
            System.out.print("Enter the per day price (INR): ");
            double perDayPrice = scanner.nextDouble();
            scanner.nextLine(); // Consume the newline character
            addCamera(new Camera(brand, model, perDayPrice));
            break;
        case 'b':
            System.out.print("Enter the camera ID to remove: ");
            int cameraId = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character
            removeCamera(cameraId);
            break;
        case 'c':
            viewMyCameras();
            break;
        case 'd':
            // Go to previous menu
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
    }
    break;
case 2:
    System.out.print("Enter the camera ID you want to rent: ");
    int rentCameraId = scanner.nextInt();
    scanner.nextLine(); // Consume the newline character
```

```
            rentCamera(rentCameraId);
            break;
        case 3:
            viewAllCameras();
            break;
        case 4:
            System.out.println("\nMy Wallet Menu:");
            System.out.println("a. My Wallet");
            System.out.println("b. Add Money");
            System.out.println("c. Go to Previous Menu");
            System.out.print("Enter your choice: ");
            char walletChoice = scanner.nextLine().charAt(0);
            switch (walletChoice) {
                case 'a':
                    viewWalletBalance();
                    break;
                case 'b':
                    System.out.print("Enter the amount to deposit (INR): ");
                    double depositAmount = scanner.nextDouble();
                    scanner.nextLine(); // Consume the newline character
                    depositToWallet(depositAmount);
                    break;
                case 'c':
                    // Go to previous menu
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
            break;
        case 5:
            System.out.println("Exiting the application. Goodbye!");
```

```
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (choice != 5);
  }
}

public class CameraRentalApp {
  public static void main(String[] args) {
    CameraRental rentalApp = new CameraRental();

    // Adding some sample cameras
    rentalApp.addCamera(new Camera("Canon", "EOS R5", 50.0));
    rentalApp.addCamera(new Camera("Nikon", "Z7 II", 60.0));
    rentalApp.addCamera(new Camera("Sony", "A7R IV", 55.0));

    rentalApp.run();
  }
}
```