1 MAIN CLASS

```
package Practice;
import java.util.Scanner;
import java.util.ArrayList;



public class Phase1Project {
	public static void main(String args[]) {

		System.out.println("\n**********\n");
		System.out.println("    !! WELCOME TO CAMERA RENTAL APP !!    ");
		System.out.println("**********");

		System.out.println("Please Login to continue");

		String correctPassword = "Hello";

		while(true) {


    Scanner sc = new Scanner(System.in);


    System.out.print("Enter the password: ");
```

```
    String enteredPassword = sc.nextLine();



    if (enteredPassword.equals(correctPassword)) {


        OptionSelection();

        ProjectFin p = new ProjectFin();

        p.main(args);

    }

       else {

       System.out.println("Incorrect password. Access denied.");

    }


        break;

                            }

}



        public static void OptionSelection() {


                String[] arrMenuOptions = {"1.  MY CAMERA ",

         "2.  VIEW ALL CAMERAS ",

         "3.  RENT A CAMERA ",
```

```
            "4.  MY WALLET ",
            "5.  ADD FUND TO WALLET ",
            "6.  EXIT "
             };

                    displayMenuOptions(arrMenuOptions);

                    }

            private static void displayMenuOptions(String[] arrMenuOptions) {
                    int slen = arrMenuOptions.length;
                    for (int i = 0; i<slen; i++)
                    {

                    System.out.println(arrMenuOptions[i]);
                     }
                       }

}
```

2. SUB CLASS 1

```
package Practice;
```

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Camera {
    String brand;
    String model;
    double rentalAmount;

    public Camera(String brand, String model, double rentalAmount) {
        this.brand = brand;
        this.model = model;
        this.rentalAmount = rentalAmount;
    }
}

class Wallet {
    double amount;

    public Wallet(double amount) {
        this.amount = amount;
    }

    public void addFunds(double funds) {
        amount += funds;
```

```
    }
}

public class ProjectFin {
    public  List<Camera> cameraList;
    private Wallet userWallet;
    private List<Camera> userAddedCameras;

    public ProjectFin() {
        cameraList = new ArrayList<>();
        userWallet = new Wallet(0);
        userAddedCameras = new ArrayList<>();

    }

    public void listCameras() {
        System.out.println("Available Cameras:");
        for (int i = 0; i < cameraList.size(); i++) {
            Camera camera = cameraList.get(i);

          System.out.println("Index: " + i + "Brand: " + camera.brand + ", Model: " + camera.model + ",
Rental Amount: $" + camera.rentalAmount);
        }

        //Display user added Camera
```

```
        if (!userAddedCameras.isEmpty()) {
            for (int i = 0; i < userAddedCameras.size(); i++) {
                Camera camera = userAddedCameras.get(i);
                int index = i + cameraList.size(); // Calculate the correct index for user-added cameras
                System.out.println("Index: " + index + ", Brand: " + camera.brand + ", Model: " + camera.model
+ ", Rental Amount: $" + camera.rentalAmount);
            }
        }
    }

    public void viewCameraDetails(int index) {
        Camera selectedCamera = cameraList.get(index);
        System.out.println("Camera Details:");
        System.out.println("Brand: " + selectedCamera.brand);
        System.out.println("Model: " + selectedCamera.model);
        System.out.println("Rental Amount: $" + selectedCamera.rentalAmount);
    }

    public void rentCamera(int index) {
        Camera selectedCamera = cameraList.get(index);
        if (selectedCamera.rentalAmount <= userWallet.amount) {
            System.out.println("Camera rented successfully!");
            userWallet.amount -= selectedCamera.rentalAmount;
        } else {
            System.out.println("Insufficient funds. Please add funds to your wallet.");
        }
```

```
}

public void viewWalletAmount() {
    System.out.println("Wallet Amount: $" + userWallet.amount);
}

public void addFundsToWallet(double funds) {
    userWallet.addFunds(funds);
    System.out.println("Funds added successfully. New wallet amount: $" + userWallet.amount);
}

public static void main(String[] args) {
    ProjectFin rentalApp = new ProjectFin();

    // Dummy camera data
    rentalApp.cameraList.add(new Camera("Canon", "EOS 5D Mark IV", 30.0));
    rentalApp.cameraList.add(new Camera("Nikon", "D850", 25.0));
    rentalApp.cameraList.add(new Camera("Sony", "A7 III", 20.0));

    Scanner scanner = new Scanner(System.in);


     while (true) {

       System.out.print("Enter your choice: ");
```

```
int choice = scanner.nextInt();



 switch (choice) {
    case 1:
                AddRem t = new AddRem();
                t.main(args);
       rentalApp.listCameras();
        break;
    case 2:
        System.out.print("Enter the camera index: ");
        int cameraIndex = scanner.nextInt();
        rentalApp.viewCameraDetails(cameraIndex);
        break;
    case 3:
        System.out.print("Enter the camera index to rent: ");
        int rentIndex = scanner.nextInt();
        rentalApp.rentCamera(rentIndex);
        break;
    case 4:
        rentalApp.viewWalletAmount();
        break;
    case 5:
        System.out.print("Enter the amount to add to your wallet: ");
        double fundsToAdd = scanner.nextDouble();
```

```
                rentalApp.addFundsToWallet(fundsToAdd);
                break;
            case 6:
                System.out.println("Exiting the application. Thank you!");
                System.exit(0);
            default:
                System.out.println("Invalid choice. Please try again.");
                break;
        }
    }

  }
}
```

3. SUB CLASS 3

```
package Practice;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
public class AddRem extends  ProjectFin {

        public   void viewCameraDetails (int index) {
```

```
// Add options for Add Camera and Remove Camera
System.out.println("\n Additional Options:");
System.out.println("1. Add Camera");
System.out.println("2. Remove Camera");
System.out.println("3. Go Back");

System.out.print("Enter your choice: ");
Scanner scanner = new Scanner(System.in);
int choice = scanner.nextInt();



switch (choice) {
   case 1:
      addCamera();
      break;
   case 2:
      removeCamera(index);
      break;

   case 3:
        viewCamera();
        break;
   case 4:
        viewCameraDetails( index);
      break;
```

```
                default:
                    System.out.println("Invalid choice. Going back to the main menu.");
                }
            }

        private void viewCamera() {
                    for (int i = 0; i < cameraList.size(); i++) {
        Camera camera = cameraList.get(i);

       System.out.println("Index: " + i + "Brand: " + camera.brand + ", Model: " + camera.model + ",
Rental Amount: $" + camera.rentalAmount);
    }

        }

        // Method to add a new camera to the list
        private  void addCamera() {
                    Scanner scanner = new Scanner(System.in);
           System.out.print("Enter the brand of the new camera: ");
           String brand = scanner.next();

           System.out.print("Enter the model of the new camera: ");
           String model = scanner.next();

           System.out.print("Enter the per-day rental amount for the new camera: ");
           double rentalAmount = scanner.nextDouble();
```

```
    Camera newCamera = new Camera(brand, model, rentalAmount);
    cameraList.add(newCamera);
   //  userAddedCameras.add(newCamera);

    System.out.println("New camera added successfully!");
}

// Method to remove a camera from the list
private void removeCamera(int index) {
    cameraList.remove(index);
    System.out.println("Camera removed successfully!");
}

public static void main(String args[]) {
        System.out.println("1.  ADD   \n  2.  REMOVE  \n 3.  VIEW MY CAMERAS   \n  4.  GO TO PREVIOUS MENU ");
        Scanner scanner = new Scanner(System.in);
        int index = scanner.nextInt();

        AddRem a = new AddRem();
        a.viewCameraDetails( index);
```

```
}

}
```