

Outputs : Camera Rental System

The screenshot shows the Eclipse IDE interface. The top toolbar includes icons for File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the toolbar, the 'Problems' tab is active, showing a warning for 'CameraRentalApp (1) [Java Application]' at 'C:\Users\home\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot'. The main editor displays the 'CameraRentalApp.java' file, which contains a Java program for a camera rental system. The code includes a main menu, a login function, and a list of cameras with their specifications and prices. The console output shows the program's execution, including the main menu, login prompt, and the successful addition of a camera to the list.

```

eclipse-workspace - PhaseEndProject/src/Project/CameraRentalApp.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

<terminated> CameraRentalApp (1) [Java Application] C:\Users\home\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot

[WELCOME TO CAMERA RENTAL APP]

Please login to continue
Enter your username: Prasad
Enter your password: Prasad78
=== MAIN MENU ===

1. MY CAMERA
2. RENT A CAMERA
3. VIEW ALL CAMERAS
4. MY WALLET
5.EXIT
Enter your choice: 1
=== SUB MENU ===

1. ADD
2. REMOVE
3. VIEW MY CAMERAS
4. GO TO PREVIOUS MENU
Enter your choice: 1
ENTER THE CAMERA BRAND: Leica
ENTER THE MODEL: Q2
ENTER THE PER DAY PRICE (ZNR): 3500
YOUR CAMERA HAS BEEN SUCCESSFULLY ADDED TO THE LIST.
=== SUB MENU ===

1. ADD
2. REMOVE
3. VIEW MY CAMERAS
4. GO TO PREVIOUS MENU

CameraRentalApp.java
// All rights reserved.
"dialogue");
", "World12");
", "Reddy567");
v", "Man1234");

1, "Sony", "4k", 15000));
2, "Kodak", "panaroma", 5000));
3, "Panasonic", "UHD", 6000));
4, "Samsung", "HD", 3000));
5, "Nikon", "Z5", 5000));
6, "Canon", "Portrait", 10000));
7, "Sigma", "Uhd56", 7000));

ner(System.in);
+ + + + +
ME TO CAMERA RENTAL APP");
+ + + + +
e login to continue");
ur username: ";
nextLine();
ur password: ";
nextLine();

.equals(username) && user.getPassword().equals(password)

;

```

[illegible]

The screenshot shows the Eclipse IDE interface. The left-hand pane contains the 'Project Explorer' and 'Console' tabs. The 'Project Explorer' shows a project named 'CameraRentalApp' with a package 'org.openjdk.hotspot'. The 'Console' tab displays the output of the application, showing a menu with options 1 to 5, and a login prompt. The right-hand pane shows the source code for 'CameraRentalApp.java'. The code includes a main method with a loop for menu navigation and a login method.

```

1. MY CAMERA
2. RENT A CAMERA
3. VIEW ALL CAMERAS
4. MY WALLET
5.EXIT
Enter your choice: 4
=== MAIN MENU ===

1. MY CAMERA
2. RENT A CAMERA
3. VIEW ALL CAMERAS
4. MY WALLET
5.EXIT
Enter your choice: 4
YOUR WALLET BALANCE IS INR.39000.0
DO YOU WANT TO DEPOSIT AMOUNT TO YOUR WALLET ?(yes/no): yes
ENTER THE AMOUNT TO DEPOSIT(INR)- 7500
Deposit successful.
YOUR WALLET BALANCE UPDATED SUCCESSFULLY.CURRENT WALLET BALANCE-INR 46500.0
=== MAIN MENU ===

1. MY CAMERA
2. RENT A CAMERA
3. VIEW ALL CAMERAS
4. MY WALLET
5.EXIT
Enter your choice: 4
YOUR WALLET BALANCE IS INR.46500.0
DO YOU WANT TO DEPOSIT AMOUNT TO YOUR WALLET ?(yes/no): no
=== MAIN MENU ===

1. MY CAMERA
2. RENT A CAMERA
3. VIEW ALL CAMERAS
4. MY WALLET
5.EXIT
Enter your choice: 5

```

```

CameraRentalApp.java
package org.openjdk.hotspot;

import java.util.Scanner;

public class CameraRentalApp {

    // Menu options
    private static final String[] menuOptions = {
        "1. MY CAMERA",
        "2. RENT A CAMERA",
        "3. VIEW ALL CAMERAS",
        "4. MY WALLET",
        "5.EXIT"
    };

    // Camera details
    private static final String[] cameraDetails = {
        "1. Sony", "4k", "15000");
        "2. Kodak", "panaroma", "5000");
        "3. Panasonic", "UHD", "6000");
        "4. Samsung", "HD", "3000");
        "5. Nikon", "Z5", "5000");
        "6. Canon", "Portrait", "10000");
        "7. Sigma", "Uhd56", "7000");
    };

    // Main method
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Display menu
        displayMenu();

        // Get user choice
        int choice = scanner.nextInt();

        // Handle choice
        switch (choice) {
            case 1:
                viewAllCameras();
                break;
            case 2:
                rentACamera();
                break;
            case 3:
                viewAllCameras();
                break;
            case 4:
                myWallet();
                break;
            case 5:
                exit();
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }

    // Login method
    public static boolean login() {
        Scanner scanner = new Scanner(System.in);

        // Get username and password
        String username = scanner.nextLine();
        String password = scanner.nextLine();

        // Check if login is successful
        if (username.equals("admin") && password.equals("password")) {
            return true;
        } else {
            return false;
        }
    }

    // Display menu method
    private static void displayMenu() {
        for (String option : menuOptions) {
            System.out.println(option);
        }
    }

    // View all cameras method
    private static void viewAllCameras() {
        for (String detail : cameraDetails) {
            System.out.println(detail);
        }
    }

    // Rent a camera method
    private static void rentACamera() {
        Scanner scanner = new Scanner(System.in);

        // Get camera ID
        int cameraId = scanner.nextInt();

        // Get rental duration
        int duration = scanner.nextInt();

        // Calculate rental amount
        double rentalAmount = calculateRentalAmount(cameraId, duration);

        // Display rental amount
        System.out.println("Rental amount: " + rentalAmount);
    }

    // My wallet method
    private static void myWallet() {
        Scanner scanner = new Scanner(System.in);

        // Get current balance
        double balance = getBalance();

        // Display current balance
        System.out.println("Current balance: " + balance);

        // Get deposit amount
        double depositAmount = scanner.nextDouble();

        // Calculate new balance
        double newBalance = balance + depositAmount;

        // Display new balance
        System.out.println("New balance: " + newBalance);
    }

    // Exit method
    private static void exit() {
        System.out.println("Exiting the application. Goodbye!");
    }

    // Calculate rental amount method
    private static double calculateRentalAmount(int cameraId, int duration) {
        // Get camera price
        double cameraPrice = getCameraPrice(cameraId);

        // Calculate rental amount
        double rentalAmount = cameraPrice * duration;

        // Return rental amount
        return rentalAmount;
    }

    // Get camera price method
    private static double getCameraPrice(int cameraId) {
        // Get camera price from details
        double cameraPrice = cameraDetails[cameraId - 1].split(",")[2].trim().replace(")", "");

        // Return camera price
        return Double.parseDouble(cameraPrice);
    }

    // Get balance method
    private static double getBalance() {
        // Get balance from wallet
        double balance = myWallet().getBalance();

        // Return balance
        return balance;
    }
}

```