ALGORITHM : CAMERA RENTAL APPLICATION

1. Define the `Camera` class with private instance variables for cameraId, brand, model, perDayPrice, and isAvailable. Include a constructor, getters, setters, and toString method.

2. Define the `User` class with private instance variables for username, password, and walletBalance. Include a constructor, getters, setters, and toString method.

3. Define the `CameraRentalApplication` class as the main class.

4. Initialize the `users` map and `cameras` list as static variables.

5. Implement the `initializeUsers` method to add user objects to the `users` map.

6. Implement the `initializeCameras` method to add camera objects to the `cameras` list.

7. Implement the `displayLoginPage` method to display the login page menu options.

8. Implement the `login` method to prompt the user to enter their username and password, validate the credentials, and log in the user if they are valid.

9. Implement the `showMainMenu` method to display the main menu options and handle user input.

10. Implement the `handleMyCameraMenu` method to display the My Camera menu options and handle user input.

11. Implement the `addCamera` method to prompt the user to enter camera details, create a new camera object, and add it to the `cameras` list.

12. Implement the `removeCamera` method to display the list of cameras, prompt the user to enter the camera ID to remove, and remove the camera from the `cameras` list if found.

13. Implement the `viewMyCameras` method to display the cameras rented by the current user.

14. Implement the `handleRentCameraMenu` method to display the Rent a Camera menu options, prompt the user to enter the camera ID to rent, and process the rental transaction.

15. Implement the `displayAllCameras` method to display the list of all cameras.

16. Implement the `displayCameraList` method to format and print the camera details in a tabular format.

17. Implement the `handleWalletMenu` method to display the Wallet menu options, show the current wallet balance of the user, and allow adding funds to the wallet.

18. Implement the `main` method to initialize users and cameras, display the login page, and handle user input until the user chooses to quit.