**ALGORITHM**

1. START
2. Create a class called "camera" to represent a camera object with attributes like ID, brand, model, rent per day, and status.
3. In the "cameraRental" class, create two ArrayLists: "camerasList" to store all available cameras and "mycamerasList" to store user's cameras.
4. Initialize the wallet balance and ID counter.
5. Implement the main method: a. Add default cameras to the "camerasList" using the "addcameravalues" method. b. Prompt the user to enter their username and password. c. If the entered credentials match the predefined admin username and password, proceed to the next step; otherwise, repeat the authentication process. d. Display the main menu using the "optionselect" method.
6. Implement the "optionselect" method: a. Display the menu options and prompt the user to enter their choice. b. Based on the user's choice, perform the following actions:
   * If the choice is 1, enter the "MY CAMERA" section:
     + Display the sub-menu options and prompt the user to enter their choice.
     + Based on the user's choice, perform actions like adding a camera, removing a camera, or viewing the user's cameras.
     + If the choice is 4, return to the main menu.
   * If the choice is 2, enter the "RENT A CAMERA" section:
     + Display all available cameras from the "camerasList."
     + Prompt the user to select a camera ID to rent.
     + Check if the user has sufficient funds in their wallet.
     + If yes, update the camera's status to "Rented," deduct the rent amount from the wallet balance, and display a success message.
     + If no, display an error message.
     + Return to the main menu.
   * If the choice is 3, display all available cameras from the "camerasList."
   * If the choice is 4, display the wallet balance and give the option to deposit money into the wallet.
   * If the choice is 5, exit the program. c. Repeat the menu options until the user chooses to exit.
7. Implement the "addcameravalues" method to add default camera options to the "camerasList."
8. Implement the "printcameras" method to display the details of cameras stored in an ArrayList.
9. Create an instance of the "cameraRental" class in the main method and call the appropriate methods to start the program.
10. Handle any input mismatch exceptions to ensure valid user input.
11. Implement proper error messages and exception handling throughout the program to provide a user-friendly experience.
12. STOP

NOTE : - username – admin

Password – admin123

User details are constant for all instances in the program