Police Inspector Eligibility Checker

By Keerthana. H [174]

1. Problem Statement

The Indian **Police Inspector Eligibility Checker** is a Java-based application designed to determine if a candidate is eligible to become a Police Inspector in India. The program considers multiple eligibility criteria which including:

- Age
- Gender
- Height
- Educational Qualification
- Nationality
- Vision
- Written Exam Result

If a candidate fails to meet any of these mandatory criteria, they are declared as not eligible with a detailed reason.

2. Requirements

Input Requirements

- Candidate’s Name (String)
- Age (Integer)
- Gender (String: "Male" or "Female")
- Height in cm (Integer)
- Educational Qualification (String)
- Nationality (String)
- Vision (String)
- Written Exam Result (String)

Criteria and required conditions for the above are

Age : 20-35 (Male), 20-30 (Female)

Height : For male should be more than 160 cm and for Female more than 150 cm

Education: Must have a Bachelor's Degree

Nationality : Must be Indian

Vision : One eye 6/6, the other 6/9

Written Exam: Must be Passed

Output Requirements

- The system should print whether a candidate is Eligible or Not Eligible.
- If not eligible, print specific reasons for rejection.
- Ask if the user wants to check another candidate.

3. Implementation Details: Justification of Features Used

Encapsulation: Candidate details are stored in private attributes for data security.

Array List: Stores multiple candidates dynamically.

Scanner Class: Takes user input for candidate details.

String Methods: Used for case-insensitive comparisons (e.g., equalsIgnoreCase).

Exception Handling: Handles invalid number inputs for age and height.

Eligibility Checker (Nested Class): Encapsulates the eligibility logic inside Candidate class.

Looping (while loop): Ensures continuous candidate entry until the user exits.

Decision Making (if-else): Validates each condition before deciding eligibility.

Objects and Classes

This program is object-oriented, means we create objects to represent real-world entities (candidates).

The Candidate class defines the properties and behaviours of a candidate.

Each candidate entered by the user is stored as an object of this class.

Access Specifiers

The Candidate class has private attributes (name, age, gender, etc.), meaning they cannot be accessed directly from outside the class.

These attributes are only accessible through a constructor or getter methods.

Constructors

The constructor in the Candidate class initializes candidate details when an object is created.

Every time a new candidate is entered, the constructor is automatically called to assign values.

Exception Handling

Used to handle invalid inputs (like entering "twenty" instead of 20 for age).

Implemented in the getValidInteger() method using a try-catch block.

Nested Classes

The EligibilityChecker class is inside the Candidate class. This allows direct access to Candidate attributes without needing extra methods.

Static Methods and Variables

checkEligibility() is static, it can be called without creating an object. The main() method is also static, since Java needs it to run the program without creating an object first.

Collections (Dynamic Data Management)

The program uses an ArrayList<Candidate> to store multiple candidates dynamically. Instead of using a fixed-size array, ArrayList grows as more candidates are added.

4. Classes and Main Functionalities

Class Candidate: Stores candidate details. Implements Encapsulation (private attributes + constructor).
Contains:

- Attributes (name, age, gender, height, education, nationality, vision, exam result).
- Constructor to initialize the object.

Nested Class (Eligibility Checker): Checks eligibility based on conditions. Uses Array List to store rejection reasons. Prints eligibility result and reasons. Rejects immediately if the written exam is failed.

Class (PoliceInspectorEligibilityChecker (Main Class)): Takes input from the user.
Calls Eligibility Checker to validate the candidate. Uses Array List to store multiple candidates. Uses Exception Handling for integer inputs

Candidate Class Main Functionalities

- public String getName() { return name; }

- public int getAge() { return age; }
- public String getGender() { return gender; }
- public int getHeight() { return height; }
- public String getEducation() { return education; }
- public String getNationality() { return nationality; }
- public String getVision() { return vision; }
- public String getWrittenExam() { return writtenExam; }

Eligibility Checker Main Functionalities

- public static boolean checkEligibility(Candidate candidate) { ... }

  Verifies age, height, nationality, vision, education, and written exam result.

  If the written exam result is "Failed", the candidate is immediately disqualified.

  Stores reasons for ineligibility and prints them.

- private static void printResult(Candidate candidate, boolean eligible, List<String> reasons) { ... }

  Displays eligibility status and rejection reasons.

Main Class Functionalities (PoliceInspectorEligibilityChecker)

- public static void main(String[] args) { ... }

  Takes candidate details as input.

  Validates integer inputs for age and height using exception handling.

  Calls checkEligibility() to determine the result.

  Asks if the user wants to check another candidate.

- private static int getValidInteger(Scanner scanner, String message) { ... }

  Ensures only valid numbers are accepted for age and height.

  Loops until a correct input is provided.

Code

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

```
import java.util.Scanner;


// Candidate class
class Candidate {
    private String name;
    private int age;
    private String gender;
    private int height;
    private String education;
    private String nationality;
    private String vision;
    private String writtenExam;
    // Constructor
public Candidate(String name, int age, String gender, int height, String education, String
nationality, String vision, String writtenExam) {
        this.name = name;
        this.age = age;
        this.gender = gender;
        this.height = height;
        this.education = education;
        this.nationality = nationality;
        this.vision = vision;
        this.writtenExam = writtenExam;
    }
public String getName() {
return name;
}


    // Eligibility Checker
static class EligibilityChecker {
public static boolean checkEligibility(Candidate candidate) {
```

```
List<String> reasons = new ArrayList<>();
boolean eligible = true;

// Written Exam Check
if (!candidate.writtenExam.equalsIgnoreCase("Pass")) {
eligible = false;
reasons.add("- Candidate must pass the written exam.");
}

// Educational Qualification Check
if (!candidate.education.equalsIgnoreCase("Bachelor's degree")) {
eligible = false;
reasons.add("- Must have a Bachelor's degree.");
}

//Age Check
if (candidate.gender.equalsIgnoreCase("Male") && (candidate.age < 20 || candidate.age >
35)) {
eligible = false;
reasons.add("- Age must be between 20 and 35 years for men.");
} else if (candidate.gender.equalsIgnoreCase("Female") && (candidate.age < 20 ||
candidate.age > 30)) {
eligible = false;
reasons.add("- Age must be between 20 and 30 years for women.");
}

// Height Check
if (candidate.gender.equalsIgnoreCase("Male")) {
if (candidate.height < 160) {
eligible = false;
reasons.add("- Height must be at least 160 cm for men. ");
```

```
}
} else if (candidate.gender.equalsIgnoreCase("Female")) {
if (candidate.height < 150) {
eligible = false;
reasons.add("- Height must be at least 150 cm for women.");
 }}
// Nationality Check
if (!candidate.nationality.equalsIgnoreCase("Indian")) {
eligible = false;
reasons.add("- Must be an Indian citizen.");
}
// Vision Check
if (!candidate.vision.equals("6/6") && !candidate.vision.equals("6/9")) {
eligible = false;
reasons.add("- Vision must be 6/6 in one eye and 6/9 in the other eye.");
}
// Print result
System.out.println("\nEligibility Result for " + candidate.getName() + ":");
if (eligible) {
System.out.println("Congratulations! " + candidate.getName() + " is eligible to become a
Police Inspector.");
} else {
System.out.println("Sorry, " + candidate.getName() + " is not eligible due to the following
reason(s):");
for (String reason : reasons) {
System.out.println(reason);
}}
 return eligible;
    }}}
// Main Class
public class PoliceInspectorEligibilityChecker12 {
```

```
private static List<Candidate> candidates = new ArrayList<>();
public static void addCandidate(Candidate candidate) {
candidates.add(candidate);
}
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
boolean continueInput = true;
while (continueInput) {
System.out.print("\nEnter Candidate's Name: ");
String name = scanner.nextLine();
int age = 0;
 while (true) {
  System.out.print("Enter Age: ");
   try {
  age = Integer.parseInt(scanner.nextLine());
  break;
 } catch (NumberFormatException e) {
  System.out.println("Invalid input! Please enter a valid integer for age.");
    }}
System.out.print("Enter Gender (Male/Female): ");
String gender = scanner.nextLine();
int height = 0;
while (true) {
System.out.print("Enter Height in cm: ");
 try {
height = Integer.parseInt(scanner.nextLine());
break;
        } catch (NumberFormatException e) {
        System.out.println("Invalid input! Please enter a valid integer for height.");
        }
```

```
        }
        System.out.print("Enter Educational Qualification: ");
        String education = scanner.nextLine();

        System.out.print("Enter Nationality: ");
        String nationality = scanner.nextLine();

        System.out.print("Enter Vision (6/6 or 6/9): ");
        String vision = scanner.nextLine();

        System.out.print("Enter Exam Result (Pass/Fail): ");
        String writtenExam = scanner.nextLine();
// Create candidate object and add to collection
Candidate candidate = new Candidate(name, age, gender, height, education, nationality,
vision, writtenExam);
        addCandidate(candidate);
        // Check eligibility
        Candidate.EligibilityChecker.checkEligibility(candidate);
        // Ask if user wants to add another candidate
        System.out.print("\nDo you want to check another candidate? (yes/no): ");
        String response = scanner.nextLine();
        if (!response.equalsIgnoreCase("yes")) {
            continueInput = false;
        }
       }
System.out.println("\nProgram Terminated. Checked " + candidates.size() + " candidates.");
scanner.close();
}}
```

```
C:\MyJava>javac PoliceInspectorEligibilityChecker12.java

C:\MyJava>java PoliceInspectorEligibilityChecker12

Enter Candidate's Name: Arjun
Enter Age: 33
Enter Gender (Male/Female): Male
Enter Height in cm: 170
Enter Educational Qualification: Bachelor's Degree
Enter Nationality: Indian
Enter Vision (6/6 or 6/9): 6/6
Enter Exam Result (Pass/Fail): Pass

Eligibility Result for Arjun:
Congratulations! Arjun is eligible to become a Police Inspector.

Do you want to check another candidate? (yes/no): yes

Enter Candidate's Name: Maya
Enter Age: 25
Enter Gender (Male/Female): Female
Enter Height in cm: 159
Enter Educational Qualification: Bachelor's Degree
Enter Nationality: Indian
Enter Vision (6/6 or 6/9): 6/6
Enter Exam Result (Pass/Fail): Pass

Eligibility Result for Maya:
Congratulations! Maya is eligible to become a Police Inspector.

Do you want to check another candidate? (yes/no): yes
```

```
Do you want to check another candidate? (yes/no): yes

Enter Candidate's Name: Ajay
Enter Age: Male
Invalid input! Please enter a valid integer for age.
Enter Age: 28
Enter Gender (Male/Female): Male
Enter Height in cm: 168
Enter Educational Qualification: Bachelor's Degree
Enter Nationality: Indian
Enter Vision (6/6 or 6/9): 6/9
Enter Exam Result (Pass/Fail): Fail

Eligibility Result for Ajay:
Sorry, Ajay is not eligible due to the following reason(s):
- Candidate must pass the written exam.

Do you want to check another candidate? (yes/no): yes

Enter Candidate's Name: Mamatha
Enter Age: 28
Enter Gender (Male/Female): Female
Enter Height in cm: 162
Enter Educational Qualification: Bachelor's Degree
Enter Nationality: Indian
Enter Vision (6/6 or 6/9): 6/9
Enter Exam Result (Pass/Fail): Pass

Eligibility Result for Mamatha:
Congratulations! Mamatha is eligible to become a Police Inspector.

Do you want to check another candidate? (yes/no): no

Program Terminated. Checked 4 candidates.
```

```
Enter Candidate's Name: Neha
Enter Age: twenty five
Invalid input! Please enter data correctly.

Enter Candidate's Name: Neha
Enter Age: 27
Enter Gender (Male/Female): Female
Enter Height in cm: 170
Enter Educational Qualification: Bachelor's degree
Enter Nationality: Indian
Enter Vision (6/6 or 6/9): 6/9

Eligibility Result for Neha:
Congratulations! Neha is eligible to become a Police Inspector.

Do you want to check another candidate? (yes/no): no
```