```java
package hw2;

import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class ScholarshipTest {

    /**
     * initial values for variables so all test cases are given the same values except for what we are testing
     */
    @BeforeEach
    void setUp() {
        Scholarship.age = 18;
        Scholarship.resident = false;
        Scholarship.partTimeWorker = false;
        Scholarship.paidStateTax = false;
        Scholarship.volunteered = false;
        Scholarship.houseHoldIncome = 0;
    }

    /**
     * Test case 1: If the user's age is less than 18 years old, they are not eligible for the scholarship.
     */
    @Test
    void smallAge() {
        setUp();
        Scholarship.age = 1;
        assertEquals(0, Scholarship.checkEligibility(Scholarship.age, Scholarship.resident, Scholarship.partTimeWorker, Scholarship.paidStateTax, Scholarship.volunteered, Scholarship.houseHoldIncome));
    }

    /**
```

```
33      * Test case 2: If the user's age is greater than
   24 years old, they are not eligible for the
   scholarship.
34      */
35     @Test
36     void largeAge() {
37         setUp();
38         Scholarship.age = 25;
39         assertEquals(0, Scholarship.checkEligibility(
   Scholarship.age, Scholarship.resident, Scholarship.
   partTimeWorker, Scholarship.paidStateTax, Scholarship.
   volunteered, Scholarship.houseHoldIncome));
40     }
41
42     /**
43      * Test case 3:  If the user's age is within the
   age limit and they have lived in California for the
   last two years, then they meet the CA residency
   requirements. This makes them eligible for the
   scholarship.
44      */
45     @Test
46     void californiaResident() {
47         setUp();
48         Scholarship.resident = true;
49         assertEquals(1, Scholarship.checkEligibility(
   Scholarship.age, Scholarship.resident, Scholarship.
   partTimeWorker, Scholarship.paidStateTax, Scholarship.
   volunteered, Scholarship.houseHoldIncome));
50     }
51
52     /**
53      * Test case 4:  If the user's age is within the
   age limit and they have worked in California (part-
   time or full time) at least for six months, then they
   meet the CA residency requirements. This makes them
   eligible for the scholarship.
54      */
```

```
55      @Test
56      void californiaWorker() {
57          setUp();
58          Scholarship.partTimeWorker = true;
59          assertEquals(1, Scholarship.checkEligibility(
    Scholarship.age, Scholarship.resident, Scholarship.
    partTimeWorker, Scholarship.paidStateTax, Scholarship.
    volunteered, Scholarship.houseHoldIncome));
60      }
61
62      /**
63       * Test case 5:  If the user's age is within the
    age limit and the student's parents have lived in
    California for at least one year, then they meet the
    CA residency requirements. This makes them eligible
    for the scholarship.
64       */
65      @Test
66      void californiaTax() {
67          setUp();
68          Scholarship.paidStateTax = true;
69          assertEquals(1, Scholarship.checkEligibility(
    Scholarship.age, Scholarship.resident, Scholarship.
    partTimeWorker, Scholarship.paidStateTax, Scholarship.
    volunteered, Scholarship.houseHoldIncome));
70
71      }
72
73      /**
74       * Test case 6:  If the user's age is within the
    age limit and the student has volunteered for a public
     cause in California and can show proof of it, then
    they meet the CA residency requirements. This makes
    them eligible for the scholarship.
75       */
76      @Test
77      void californiaVolunteer()
78      {
```

```
 79          setUp();
 80          Scholarship.volunteered = true;
 81          assertEquals(1, Scholarship.checkEligibility(
    Scholarship.age, Scholarship.resident, Scholarship.
    partTimeWorker, Scholarship.paidStateTax, Scholarship
    .volunteered, Scholarship.houseHoldIncome));
 82      }
 83
 84      /**
 85       * Test case 7: If the user's age is within the
    age limit and the student does not meet any CA
    residency requirements, yet they have an income less
    than $5,000 then they qualify for Dean for
    consideration.
 86       */
 87      @Test
 88      void noResidentYesIncome()
 89      {
 90          setUp();
 91          assertEquals(-1, Scholarship.checkEligibility
    (Scholarship.age, Scholarship.resident, Scholarship.
    partTimeWorker, Scholarship.paidStateTax, Scholarship
    .volunteered, Scholarship.houseHoldIncome));
 92      }
 93
 94      /**
 95       * Test case 8:  If the user's age is within the
    age limit, the student does not meet any CA residency
     requirements, and they have an income more than $5,
    000 then they are not eligible for the scholarship or
     for dean consideration
 96       */
 97      @Test
 98      void noResidentNoIncome()
 99      {
100          setUp();
101          Scholarship.houseHoldIncome = 123456789;
102          assertEquals(0, Scholarship.checkEligibility(
```

```
102 Scholarship.age, Scholarship.resident, Scholarship.
    partTimeWorker, Scholarship.paidStateTax, Scholarship
    .volunteered, Scholarship.houseHoldIncome));
103     }
104 }
```

```java
package hw2;
/*
  Authors: Elizabeth Hillman, Camellia Bazargan, Vy Nguyen, Jagjit Singh
  Date: 10/01/22
  Assignment: Scholarship - Control Flow
  Problem Statement: Determine if a student is eligible for a scholarship or for dean consideration
  based on their age, where they live, their work history, taxes, volunteering history, and income

 */

import java.util.Scanner;

public class Scholarship {
    //variables used to determine the students eligibility
    static int age = 0;
    static boolean resident = false;
    static boolean partTimeWorker = false;
    static boolean paidStateTax = false;
    static boolean volunteered = false;
    static int houseHoldIncome = 0;

    /**
     * This method will call checkEligibility to determine the student's eligibility status
     */
    public static void main(String[] args)
    {
        //scanner to get user info
        Scanner sc = new Scanner(System.in);

        //used to get the users age
        System.out.println("Enter your age");
        age = sc.nextInt();

```

```
34          //used to verify the user has lived in CA for
    2 years
35          System.out.println("Have you lived in
    california for last 2 years: Y/N");
36          resident = charToBool(sc.next().charAt(0));
37
38          //used to verify the users work history
39          System.out.println("have you worked part time
    or full time for at least 6 months");
40          partTimeWorker = charToBool(sc.next().charAt(0
    ));
41
42          //used to verify the user's parents tax info
43          System.out.println("have your parents paid CA
    state tax and lived in CA for 1 year");
44          paidStateTax = charToBool(sc.next().charAt(0
    ));
45
46          //used to verify the users volunteering
    history
47          System.out.println("have you volunteered for a
     public cause in CA and have proof");
48          volunteered = charToBool(sc.next().charAt(0));
49
50          //used to very the users income
51          System.out.println("Enter your HouseHold
    Income");
52          houseHoldIncome = sc.nextInt();
53
54          System.out.println(checkEligibility(age,
    resident, partTimeWorker, paidStateTax, volunteered,
    houseHoldIncome));
55      }
56
57      /**
58       * this method will be used to determine the users
     eligibility for the scholarship based on their input
59       * prints 0 if they aren't eligible, 1 if they are
```

```
59  or for dean for consideration
60       */
61      public static int checkEligibility(int userAge,
    boolean isResident, boolean isWorker, boolean
    paidTaxes, boolean hasVolunteered, int income)
62      {
63
64          //if they do not meet age requirements, they
    will not be eligible and code will end
65          if (userAge < 18 || userAge > 24)
66          {
67              return 0;
68          }
69
70          //checks if the user meets at least one CA
    residency conditions
71          if (isResident || isWorker || paidTaxes ||
    hasVolunteered) {
72              return 1;
73          }
74
75          //if they do not meet residency conditions but
     meet requirements then they will be up for dean
    consideration
76          if (income < 5000) {
77              System.out.println("Dean for consideration
    ");
78              return -1;
79          }
80          //if they don't meet residency conditions and
    also have too high of an income they will not be
    eligible
81          return 0;
82      }
83
84      /**
85       * this method will alter the users text input to
    be saved as a boolean value
```

```
86      * @param c - user input (y or n)
87      * @return - boolean value based on input or an
    exception if the user input is incorrect
88      */
89     public static boolean charToBool(char c) {
90         return switch (c) {
91             case 'y' -> true;
92             case 'n' -> false;
93             default -> throw new
    IllegalArgumentException("Must be either 'y' or 'n'."
    );
94         };
95     }
96 }
97
```