CMPE-187

Professor Ishie Eswar

Group 2

Feb 28, 2026

CONTROL FLOW - SCHOLARSHIP ELIGIBILITY

I. Source Code

```
/**
 * Scholarship Eligibility Checker
 * Determines if a student is eligible for a scholarship based on specific criteria
 */
public class ScholarshipEligibility {

    /**
     * Checks scholarship eligibility based on student criteria
     *
     * @param age Student's age
     * @param caResidency2Years Student lived in CA for last 2 years
     * @param caWork6Months Student worked in CA for 6+ months
     * @param parentsCATax Parents paid CA tax for 1+ years
     * @param volunteeredCA Student volunteered in CA with proof
     * @param householdIncome Annual household income
     * @returN ENUM: ELIGIBLE, NOT_ELIGIBLE, DEFER_TO_DEAN
     */
    public static String evaluateEligibility(
            int age,
            boolean caResidency2Years,
            boolean caWork6Months,
            boolean parentsCATax,
            boolean volunteeredCA,
            double householdIncome) {

        // Criterion 1: Age must be between 18 and 24 (inclusive)
        if (age < 18 || age > 24) {
            return "NOT ELIGIBLE";
        }

        // Criterion 2: Check CA Residency (at least one must be true)
        boolean residencyMet = caResidency2Years || caWork6Months ||
                               parentsCATax || volunteeredCA;

        if (residencyMet) {
            return "ELIGIBLE";
        }

        // Criterion 3: Dean's Consideration
        if (householdIncome < 5000) {
            return "DEAN FOR CONSIDERATION";
        }

        return "NOT ELIGIBLE";
    }

    public static void main(String[] args) {
        /**
         * Test Case 1: "Dean consideration"
         * Age (22) check FALSE
         * Residency not met: ALL FALSE
         * Income (3000 < 5000): True
         * Return: Defer to Dean
         */
        String test_result_consideration =
            evaluateEligibility(22, false,
                false, false, false, 3000);

        /**
         * Test Case 2: Age Fail
         * Age (25) check False
         * The other fields are randomly input
         * Return: Not eligible
         */
```

```java
        String test_result_false =
            evaluateEligibility(25, true,
                true, true, false, 6000);

        /**
         * Test casse 3: Residency met
         * Age (19) check True
         * Residency met: more than one condition for residency is true
         * Income is not evaluated
         * Return: Eligible
         */
        String test_result_true =
            evaluateEligibility(19, true,
                true, true, false, 2500);

        /**
         * Test Case 4: Not Eligible
         * Age (20) check True
         * Residency not met: all conditions are False
         * Income (6000 > 5000): False
         * Return Not Eligible
         */
        String test_result_false2 = ScholarshipEligibility.evaluateEligibility(20, false,
                false, false, false, 6000
);

        // Print the result
        System.out.println(test_result_consideration);
        System.out.println(test_result_false);
        System.out.println(test_result_true);
        System.out.println(test_result_false2);
    }
}

```

II. Control Flow

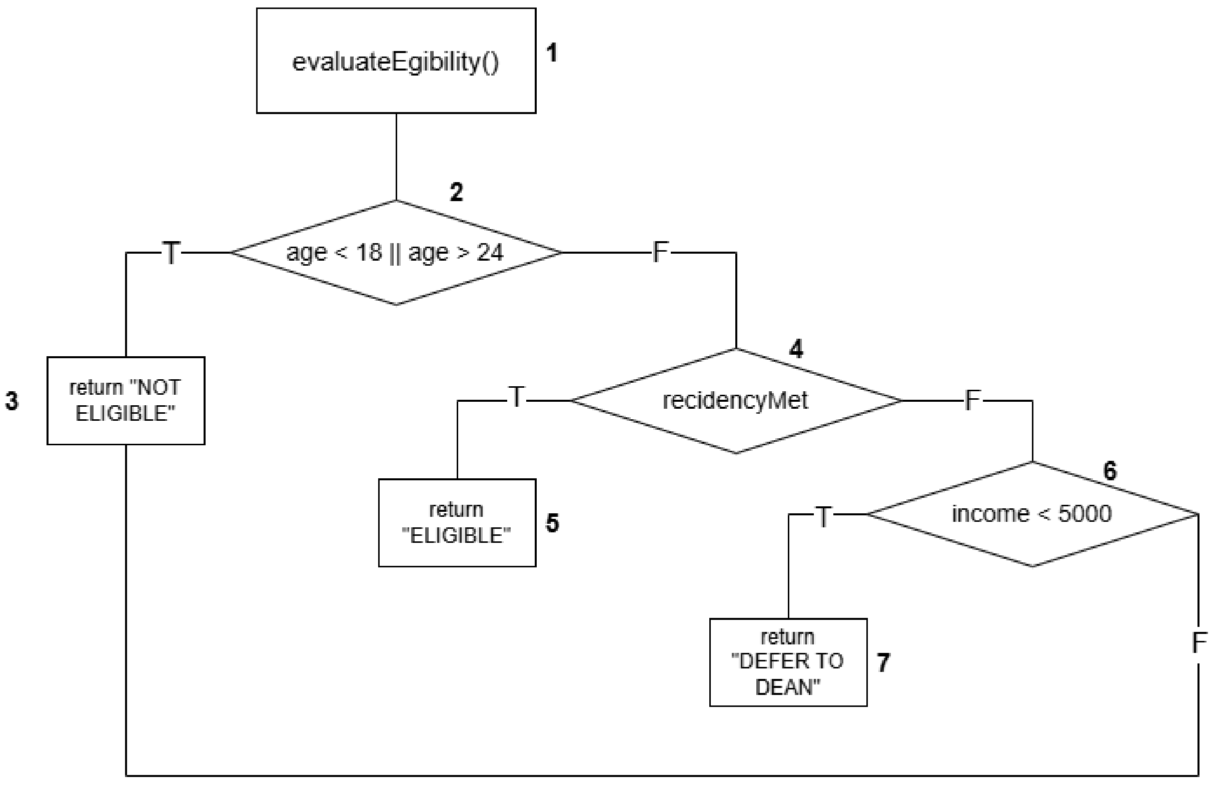

Diagram 1. Control Flow Diagram of the evaluateEgibility()

- Path 1 - Age Check Fails: 1 - 2(T) - 3
- Path 2 - Age Check Passes, ResidencyMet is True: 1 - 2(F) - 4(T) - 5
- Path 3 - Age Check Passes, ResidencyMet is False, Income >= 5000:

  1 - 2(F) - 4(F) - 6 (F) - 3
- Path 4 - Age Check Passes, ResidencyMet is False, Income < 5000:

  1 - 2(F) - 4(F) - 6 (T) - 7

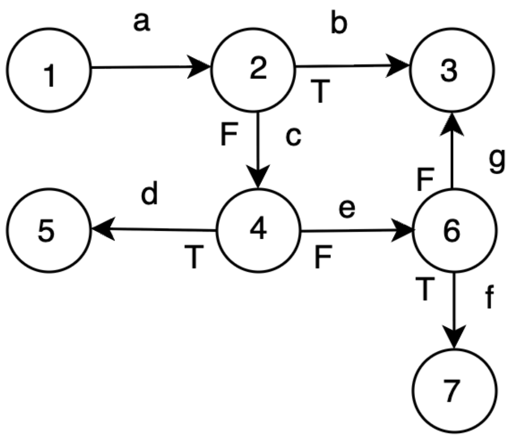

Diagram 2. Branch for Control Flow Graph

III. Test Cases

We need to execute all return statements:

1. return "NOT ELIGIBLE" (age fail)
2. return "ELIGIBLE"
3. return "DEAN FOR CONSIDERATION"
4. Final return "NOT ELIGIBLE”

Statement Coverage = 100% with 4 test cases

We cannot do it with fewer because there are 4 different return statements.

Decisions in program:

1. Age check → True / False
2. Residency check → True / False
3. Income check → True / False

That’s 6 branches total.

Test Case	Input	Expected Output	Actual Output
TC1	Age = 25	“NOT ELIGIBLE”	“NOT ELIGIBLE”
TC2	Age = 20, Residency = true	“ELIGIBLE”	“ELIGIBLE”
TC3	Age = 22, Residency = false, Income = 3000	“DEAN FOR CONSIDERATION”	“DEAN FOR CONSIDERATION”
TC4	Age = 20, Residency = false, Income = 6000	“NOT ELIGIBLE”	“NOT ELIGIBLE”

Table 1. Test Cases for the evaluation of Financial Aid

To cover both True and False outcomes for each decision, we need: Minimum 4 Test Cases

The same 4 test cases above achieve full branch coverage.

IV. Coverage Table

PATHS	DECISIONS			PROCESS LINKS						
	2	*4*	*6*	*a*	*b*	*c*	*d*	*e*	*f*	*g*
ab	T			*	*					
acd	F	T		*		*	*			
acef	F	F	T	*		*		*	*	
aceg	F	F	F	*		*		*		*

Table 2. Branch Coverage Table