

## Equivalence Class Test

To implement the test, we will follow these steps:

1. **Identify Equivalence Classes:**
  - Determine the valid and invalid input ranges for each variable.
2. **Design Test Cases:**
  - Create test cases that cover each equivalence class using different methods such as weak normal, strong normal, weak robust, and strong robust ECT.
3. **Implement Test Cases in Java:**
  - Write test cases in Java using JUnit.

### Step 1: Identify Equivalence Classes

For this example, let's assume we're validating the signup method of the UserManager class in the EduTrack application. The method takes the following parameters:

- username: String
- password: String
- confirmPassword: String
- email: String
- fullName: String

### Step 2: Design Test Cases

#### Weak Normal ECT

1. **Valid Inputs**
  - Case 1: All inputs valid.

#### Valid Cases Table (Weak Normal ECT)

Case	Username	Password	Confirm Password	Email	Full Name	Expected Outcome
1	validUser	ValidPass1!	ValidPass1!	test@example.com	John Doe	true

#### Strong Normal ECT

1. **Valid Inputs**
  - Case 1: All inputs valid.
  - Case 2: Valid username, password mismatch.
  - Case 3: Valid password, invalid email.

#### Valid Cases Table (Strong Normal ECT)

Case	Username	Password	Confirm Password	Email	Full Name	Expected Outcome
1	validUser	ValidPass1!	ValidPass1!	test@example.com	John Doe	true
2	validUser	ValidPass1!	WrongPass1!	test@example.com	John Doe	false

3	validUser	ValidPass1!	ValidPass1!	invalidEmail	John Doe	false
---	-----------	-------------	-------------	--------------	----------	-------

**Weak Robust ECT****1. Invalid Inputs**

- Case 1: Invalid username.
- Case 2: Invalid password.
- Case 3: Invalid email.

**Invalid Cases Table (Weak Robust ECT)**

Case	Username	Password	Confirm Password	Email	Full Name	Expected Outcome
1	u	ValidPass1!	ValidPass1!	test@example.com	John Doe	false
2	validUser	pass	pass	test@example.com	John Doe	false
3	validUser	ValidPass1!	ValidPass1!	invalidEmail	John Doe	false

**Strong Robust ECT****1. Combination of valid and invalid inputs**

- Case 1: Invalid username, valid others.
- Case 2: Valid username, invalid password.
- Case 3: Valid username, valid password, invalid email.

**Invalid Cases Table (Strong Robust ECT)**

Case	Username	Password	Confirm Password	Email	Full Name	Expected Outcome
1	u	ValidPass1!	ValidPass1!	test@example.com	John Doe	false
2	validUser	pass	pass	test@example.com	John Doe	false
3	validUser	ValidPass1!	ValidPass1!	invalidEmail	John Doe	false

**Step 3: Implement Test Cases in Java**

Here is an implementation of the test cases using JUnit:

```
package PathTesting;
```

```
import Proj_375_Classes.UserManager;
```

```
import static org.junit.jupiter.api.Assertions.*;
```

```
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

public class EquivalenceClassTest {

    private UserManager userManager;

    @BeforeEach
    void setUp() {
        userManager = new UserManager();
    }

    @Test
    void testWeakNormalECT() {
        // Valid Case for Weak Normal ECT
        assertTrue(userManager.signup("validUser", "ValidPass1!", "ValidPass1!",
            "test@example.com", "John Doe"));
    }

    @Test
    void testStrongNormalECT() {
        // Valid Cases for Strong Normal ECT
        assertTrue(userManager.signup("validUser", "ValidPass1!", "ValidPass1!",
            "test@example.com", "John Doe"));
        assertFalse(userManager.signup("validUser", "ValidPass1!", "WrongPass1!",
            "test@example.com", "John Doe"));
        assertFalse(userManager.signup("validUser", "ValidPass1!", "ValidPass1!", "invalidEmail",
            "John Doe"));
    }

    @Test
    void testWeakRobustECT() {
        // Invalid Cases for Weak Robust ECT
        assertFalse(userManager.signup("u", "ValidPass1!", "ValidPass1!", "test@example.com",
            "John Doe"));
        assertFalse(userManager.signup("validUser", "pass", "pass", "test@example.com", "John
            Doe"));
        assertFalse(userManager.signup("validUser", "ValidPass1!", "ValidPass1!", "invalidEmail",
            "John Doe"));
    }

    @Test
    void testStrongRobustECT() {
        // Invalid Cases for Strong Robust ECT
```

```
assertFalse(userManager.signup("u", "ValidPass1!", "ValidPass1!", "test@example.com",  
"John Doe"));  
assertFalse(userManager.signup("validUser", "pass", "pass", "test@example.com", "John  
Doe"));  
assertFalse(userManager.signup("validUser", "ValidPass1!", "ValidPass1!", "invalidEmail",  
"John Doe"));  
}  
}
```

### **Conclusion**

By following these steps, we effectively validated the EduTrack application using Equivalence Class Testing. This approach ensures that all important cases are covered, including both valid and invalid inputs. The provided Java code helps implement these test cases.