Lab 4: Introduction to Equivalence Partitioning

Objective:

To understand the concept of Equivalence Partitioning (EP) and explore different techniques for applying it in software testing to design effective test cases.

Introduction:

Equivalence Partitioning is a black-box testing technique used to divide input data into partitions that are expected to exhibit similar behavior. By testing one representative value from each partition, testers can efficiently cover a range of inputs with fewer test cases while maintaining high coverage. This lab manual introduces the fundamentals of Equivalence Partitioning and demonstrates various techniques to apply it effectively in test case design.

Equipment and Tools:

- Computer with a text editor or testing tool (e.g., Microsoft Word, Notepad, or any test management tool).
- Sample application or system specification for testing (provided in exercises).
- Pen and paper for manual calculations (optional).

Theoretical Background:

Equivalence Partitioning assumes that a system processes all inputs within a partition in the same way. If one value in a partition works correctly, other values in that partition are assumed to work similarly. Partitions can be valid (inputs that should be accepted) or invalid (inputs that should be rejected).

Key Concepts:

- Valid Partitions: Ranges of inputs that the system should process correctly.
- Invalid Partitions: Ranges of inputs that the system should reject or handle as errors.
- Boundary Values: Values at the edges of partitions, often tested separately (see Boundary Value Analysis).

Equivalence Partitioning Techniques:

The following techniques are commonly used to apply Equivalence Partitioning:

1. Range-Based Partitioning: Divide numerical input ranges into valid and invalid partitions.
2. Value-Based Partitioning: Group discrete values (e.g., categories, types) into partitions based on system behavior.
3. Condition-Based Partitioning: Partition inputs based on logical conditions or rules defined in the system.
4. List-Based Partitioning: Use predefined lists or sets (e.g., dropdown options) to create partitions.

Lab Exercises:

The following exercises will help you apply Equivalence Partitioning techniques to design test cases.

Exercise 1: Range-Based Partitioning

Scenario: A system accepts ages between 18 and 60 for a job application form. Ages outside this range are rejected.

Task:

1. Identify the valid and invalid equivalence partitions.
2. Select one test case from each partition.
3. Document the partitions and test cases in a table.

Solution Approach:

- Valid Partition: Ages 18 to 60.
- Invalid Partitions: Ages < 18, Ages > 60.
- Choose representative values: e.g., 17 (invalid), 25 (valid), 65 (invalid).

Deliverable: Complete the table below.

Partition	Type	Test Case
Ages < 18	Invalid	17
Ages 18–60	Valid	25
Ages > 60	Invalid	65

Table 1: Equivalence Partitions for Age Input

Exercise 2: Value-Based Partitioning

Scenario: A dropdown menu allows users to select a user role: Admin, Editor, or Viewer. Each role grants different permissions.

Task:

1. Identify the equivalence partitions based on the roles.
2. Select one test case for each partition.
3. Document the test cases.

Solution Approach:

- Valid Partitions: Admin, Editor, Viewer.
- Invalid Partition: Any input not in the dropdown (e.g., Guest).
- Test cases: Admin, Editor, Viewer, and an invalid input like Guest.

Deliverable: Create a test case table similar to Exercise 1.

Exercise 3: Condition-Based Partitioning

Scenario: A login system requires a password to be 8–16 characters long, contain at least one uppercase letter, one lowercase letter, and one digit.

Task:

- Identify valid and invalid partitions based on the password rules.
- Select representative test cases for each partition.
- Document the partitions and test cases.

Solution Approach:

- Valid Partition: Passwords meeting all criteria (e.g., ″Passw0rd123″).
- Invalid Partitions:
    - Passwords < 8 characters (e.g., ″Pw1″).
    - Passwords > 16 characters (e.g., ″Password123456789″).
    - Passwords missing uppercase (e.g., ″password123″).
    - Passwords missing lowercase (e.g., ″PASSWORD123″).
    - Passwords missing digits (e.g., ″Password″).

Deliverable: Document partitions and test cases in a table.

Exercise 4: List-Based Partitioning

Scenario: A form accepts country codes from a predefined list: US, UK, CA, AU. Any other code is invalid.

Task:

1. Identify the valid and invalid partitions.
2. Select one test case per partition.
3. Document the results.

Solution Approach:

- Valid Partition: US, UK, CA, AU.
- Invalid Partition: Any code not in the list (e.g., JP).
- Test cases: US, UK, CA, AU, and JP (invalid).

Deliverable: Create a test case table.

Lab Procedure:

1. Read the scenario for each exercise carefully.
2. Identify the input domain and divide it into equivalence partitions (valid and invalid).
3. Select one representative test case for each partition.
4. Document the partitions and test cases in a table format.
5. Discuss with peers or the instructor to verify your partitions.
6. If a testing tool is available, input your test cases and observe the system’s behavior.

Discussion Questions:

1. How does Equivalence Partitioning reduce the number of test cases while maintaining coverage?
2. What challenges did you face when identifying partitions for complex inputs (e.g., Exercise 3)?
3. How can Equivalence Partitioning be combined with Boundary Value Analysis for better testing?
4. In what scenarios might Equivalence Partitioning be less effective?

Conclusion:

Equivalence Partitioning is a powerful technique to optimize test case design by grouping inputs into partitions with similar behavior. By practicing the techniques covered in this lab (range-based, value-based, condition-based, and list-based partitioning), you can create efficient and effective test cases for various systems.

References:

- Myers, G. J., Sandler, C., Badgett, T. (2011). The Art of Software Testing. Wiley.
- ISTQB Foundation Level Syllabus, https://www.istqb.org.