Experiment no. 8

To design effective unit test cases using Equivalence Class Testing (ECT) to ensure comprehensive input coverage with minimal redundancy.

Case Study

Online Course Registration System

Type

Individual Assignment

Objective

To apply **Equivalence Class Testing (ECT)** to design unit test cases for an Online Course Registration System. Students will learn to ensure **comprehensive input coverage with minimal redundancy** by identifying valid and invalid partitions for different input fields.

You are required to:

1. Identification of Input Fields

Sr. No.	Input Field	Description
1	Student ID	Must be 8 digits long and numeric
2	Number of Courses	Must be between 3 and 6
3	Course Code	Format: XX999 (two letters followed by three digits)
4	Payment Mode	Credit, Debit, UPI, or NetBanking

---

2. Definition of Valid and Invalid Equivalence Classes

Input Field	Valid Equivalence Classes	Invalid Equivalence Classes
Student ID	8-digit numeric ID (e.g., 12345678)	Less than or more than 8 digits, non-numeric values

Number of Courses	3–6 inclusive	Less than 3 or greater than 6
Course Code	Two letters followed by three digits (e.g., CS101)	Missing letters/digits, extra characters, wrong format (e.g., C101, CSC101, 12345)
Payment Mode	Credit, Debit, UPI, NetBanking	Any other mode (e.g., Cash, PayPal, Crypto)

---

3. Design of Unit Test Cases (ECT Table)

Test Case ID	Input Values	Equivalence Class	Expected Output
TC01	Student ID: 12345678 Courses: 4 Code: CS101 Payment: Credit	All Valid	Registration Successful
TC02	Student ID: 1234 Courses: 4 Code: CS101 Payment: Credit	Invalid Student ID	Registration Rejected (Invalid ID)
TC03	Student ID: 12345678 Courses: 2 Code: CS101 Payment: UPI	Invalid Course Count	Registration Rejected (Less than 3 courses)
TC04	Student ID: 12345678 Courses: 7 Code: CS101 Payment: NetBanking	Invalid Course Count	Registration Rejected (Exceeds max course limit)
TC05	Student ID: 12345678 Courses: 5 Code: C101 Payment: Debit	Invalid Course Code	Registration Rejected (Invalid code format)

TC06	Student ID: 87654321 Courses: 6 Code: IT202 Payment: PayPal	Invalid Payment Mode	Registration Rejected (Invalid payment mode)
TC07	Student ID: ABCD5678 Courses: 4 Code: CS101 Payment: UPI	Invalid Student ID (non-numeric)	Registration Rejected (Invalid ID format)
TC08	Student ID: 13579246 Courses: 3 Code: MA303 Payment: NetBanking	All Valid	Registration Successful

---

4. Comparison: ECT vs Special Value Testing (SVT)

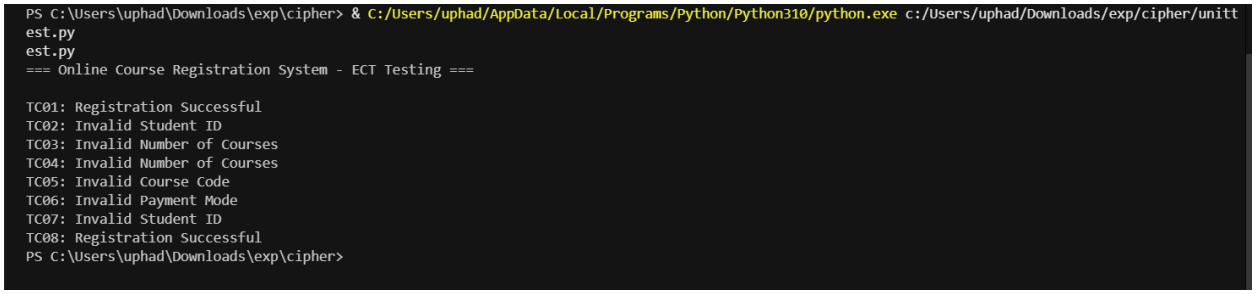
Aspect	Equivalence Class Testing (ECT)	Special Value Testing (SVT)
Purpose	Divides input domain into partitions to minimize redundancy	Focuses on boundary or special/critical input values
Coverage	Tests representative values from each valid and invalid class	Tests limit or extreme values (e.g., min/max inputs)
Example (No. of Courses)	Classes: <3 (invalid), 3–6 (valid), >6 (invalid)	Special values: 2, 3, 6, 7
Efficiency	Reduces total number of test cases	Ensures boundary correctness
Best Used When	Input domain is large and needs reduced test set	Input boundaries or critical edge values matter most

Program and Outcome:

```
def validate_registration(student_id, num_courses, course_code,
payment_mode):
    if len(student_id) != 8 or not student_id.isdigit():
        return "Invalid Student ID"
    if num_courses < 3 or num_courses > 6:
        return "Invalid Number of Courses"
    if not (len(course_code) == 5 and course_code[:2].isalpha() and
course_code[2:].isdigit()):
        return "Invalid Course Code"
    if payment_mode not in ["Credit", "Debit", "UPI", "NetBanking"]:
        return "Invalid Payment Mode"
    return "Registration Successful"

test_cases = [
    ("TC01", "12345678", 4, "CS101", "Credit"),
    ("TC02", "1234", 4, "CS101", "Credit"),
    ("TC03", "12345678", 2, "CS101", "UPI"),
    ("TC04", "12345678", 7, "CS101", "NetBanking"),
    ("TC05", "12345678", 5, "C101", "Debit"),
    ("TC06", "87654321", 6, "IT202", "PayPal"),
    ("TC07", "ABCD5678", 4, "CS101", "UPI"),
    ("TC08", "13579246", 3, "MA303", "NetBanking")
]

print("=== Online Course Registration System - ECT Testing ===\n")
for tc in test_cases:
    tc_id, sid, courses, code, payment = tc
    result = validate_registration(sid, courses, code, payment)
    print(f"{tc_id}: {result}")
```



5. Conclusion

Using **Equivalence Class Testing (ECT)**, we systematically divided the input domain of the **Online Course Registration System** into valid and invalid classes. This ensured **complete input coverage** while **reducing redundant test cases**. ECT helps in efficient and reliable test design and can be complemented with **Special Value Testing (SVT)** for boundary validation.