**PROBLEM STATEMENT:-**

**SPECIFICATION-BASED TESTING**

Suppose a software component (called a Grader component) has been implemented to automatically compute a grade in a course. A course taught at a university has two exams and a project. To pass the course with grade C a student must score at least 45 points in the Exam-1, 50 points in Exam-2, and 50 points in Project. Students pass the course with grade B if they score at least 60 points in the Exam-1, 55 points in Exam-2, and 60 points in Project. If, in addition to this, the average of the exams is at least 80 points and they score at least 70 points in Project then students are awarded a grade A. Final grade for the course are: A, B, C, and E. The Grader component accepts six inputs:

Last name First name Student # Exam-1 Exam-2

**Project Assumptions:**

1. Assume Exam-1, Exam-2, Project are integers.
2. The ranges for the exam scores are:
3. 0 <= Exam-1 <= 90
4. 0 <= Exam-2 <= 100
5. 0 <= Project <= 80
6. The maximum size of the “First name” is 15 characters and “Last name” is 20 characters.
7. Student # is a number represented as a 9-character string in the following format: AXXXXXXXX, where X is a digit.

Sample test case for the Grader component:

Test #1: Last name=Smith, First name=John, Student #=A11112222, Exam-1=57, Exam-2 = 64, Project = 75

**PROBLEM #1 : Equivalence partition testing**

Identify input conditions for the Grader component related to:

1. Last name
2. First name
3. Student #
4. Exam-1
5. Exam-2
6. Project

From the identified input conditions list equivalence valid and invalid sub-domains (classes). Based on the identified sub-domains design test cases using:

a. Strong normal equivalence testing,

b. Weak robust equivalence testing