**Assignment 2 - Equivalence Partitioning Testing for a Loan Application Validator using JUnit**

**Objective:**

To validate a complex system by partitioning input data into **equivalence classes** and writing JUnit test cases to cover representative values from each class.

**Problem Statement:**

Design a loan application validator system with the following constraints:

| **Input Field** | **Valid Range or Class** |
| --- | --- |
| Age | 21 – 60 |
| Annual Income | > 2,50,000 (INR) |
| Employment Type | "Salaried", "Self-Employed" |
| Credit Score | 300 – 850 (minimum acceptable: 650) |

You are to validate applications and return true only if **all criteria are met**.

**Java Code: LoanValidator.java**

public class LoanValidator {

public static boolean isValidApplication(int age, double income, String employmentType, int creditScore) {

return isAgeValid(age) &&

isIncomeValid(income) &&

isEmploymentTypeValid(employmentType) &&

isCreditScoreValid(creditScore);

}

public static boolean isAgeValid(int age) {

return age >= 21 && age <= 60;

}

public static boolean isIncomeValid(double income) {

return income > 250000;

}

public static boolean isEmploymentTypeValid(String employmentType) {

return "Salaried".equalsIgnoreCase(employmentType) ||

"Self-Employed".equalsIgnoreCase(employmentType);

}

public static boolean isCreditScoreValid(int creditScore) {

return creditScore >= 650 && creditScore <= 850;

}

}

**Submission Guideline:**

1. Make a solution file (Java File) and name the file name as YOUR\_NAME\_EP.java
2. Upload on following Google Drive Link

<https://drive.google.com/drive/folders/1u-nWkZ2fDIfQTUVbnrJHKBPmXB_fqMoE?usp=sharing>