**FORMAT of the data in file:**

1. Policy code
2. Pan
3. Start date
4. Period
5. Accumulated premium
6. Requested loan amount

**Validation:**

1. Policy code should be of [policy type-policy no-sum assured] format
   1. Policy number should be
      1. Non zero digit
      2. Unique number
   2. Sum assured should be digit
2. Policy type should be either FRM or NRS
3. Accumulated premium should be less than fa
4. Pan should start with FR and after FR there should be 3 digits
5. None of the field should be empty

**Status Computation:**

Eligibility Status:

|  |  |  |
| --- | --- | --- |
| type | Calculation: Loan amount | status |
| FRM | >40 PERCENT OF ACCU PREM | NELG |
| FRM | <40 percent of accu prem | ELG |
| NRS | <60 percent of accu prem | ELG |
| NRS | >60 perent accu prem and <70 percent of fa | ELG |
| NRS | Greater than 60 perent accu prem and greater than 70 percent of fa | NELG |

Policy Status:

1. **DUP**: if the policy number of two vos are same
2. **INV**: if loan amount greater than sum assured

**Net Premium Computation:**

1. If status = NELG, then net premium = accu prem
2. If status = ELG, then net premium = x + y, where:
   1. Y = (loan amount +interest) \* loan period
   2. Loan period = months between end date and sanction date
   3. X = accu prem / period

**Input parameter of the method(getInsuranceDetails):**

1. File path
2. sanction date (in string format)

**Output map structure:**

Output will be a Map<Integer, Map<String, List<policyDetailsVO >>

Integer : 1 :: ELG/NELG : LIST  
Integer : 2 :: DUP/INV : LIST

|  |  |
| --- | --- |
| **Integer** | **Inner map structure**  MAP<String, List> |
| 1 | ELG: list of **eligible** policy vos  NELG: list of **noneligible** policy vos |
| 2 | DUP: list of **duplicate** policy vos  INV: list of **invalid** policy vos |