CONCURRENT MIX CALCULATOR

REQUIREMENTS

Hardware Requirements

- ❖ Storage: Minimum 4GB RAM
- ❖ Processor: Intel Core i3
- ❖ Input Devices: Keyboard, Mouse
- ❖ Personal Computers

Software Requirements

- ❖ Windows 7/8/10/11
- ❖ Linux
- ❖ Mac Os
- ❖ MS Word

Introduction

- In civil engineering and construction materials testing, achieving the correct aggregate gradation is essential for producing high-quality concrete, asphalt, and pavement layers. A well-graded aggregate mix improves strength, durability, workability, and overall performance of the final construction material. However, aggregates used on site often come from different sources and have varying gradation characteristics. To obtain a desired final gradation, these different aggregates must be blended in suitable proportions. This process is known as **concurrent mixing**.
- A **Concurrent Mix Calculator** is a computational tool designed to simplify and accurately determine the combined gradation when two or more aggregates are mixed together. Instead of manually performing repeated calculations for each sieve size and proportion, the calculator automates the process, ensuring fast, error-free, and reliable results. By applying weighted average calculations across all sieve sizes, the system generates the final percentage passing or retained for each sieve, enabling engineers to validate whether the mix meets specific standards such as MORTH, ASTM, or project-based requirements.

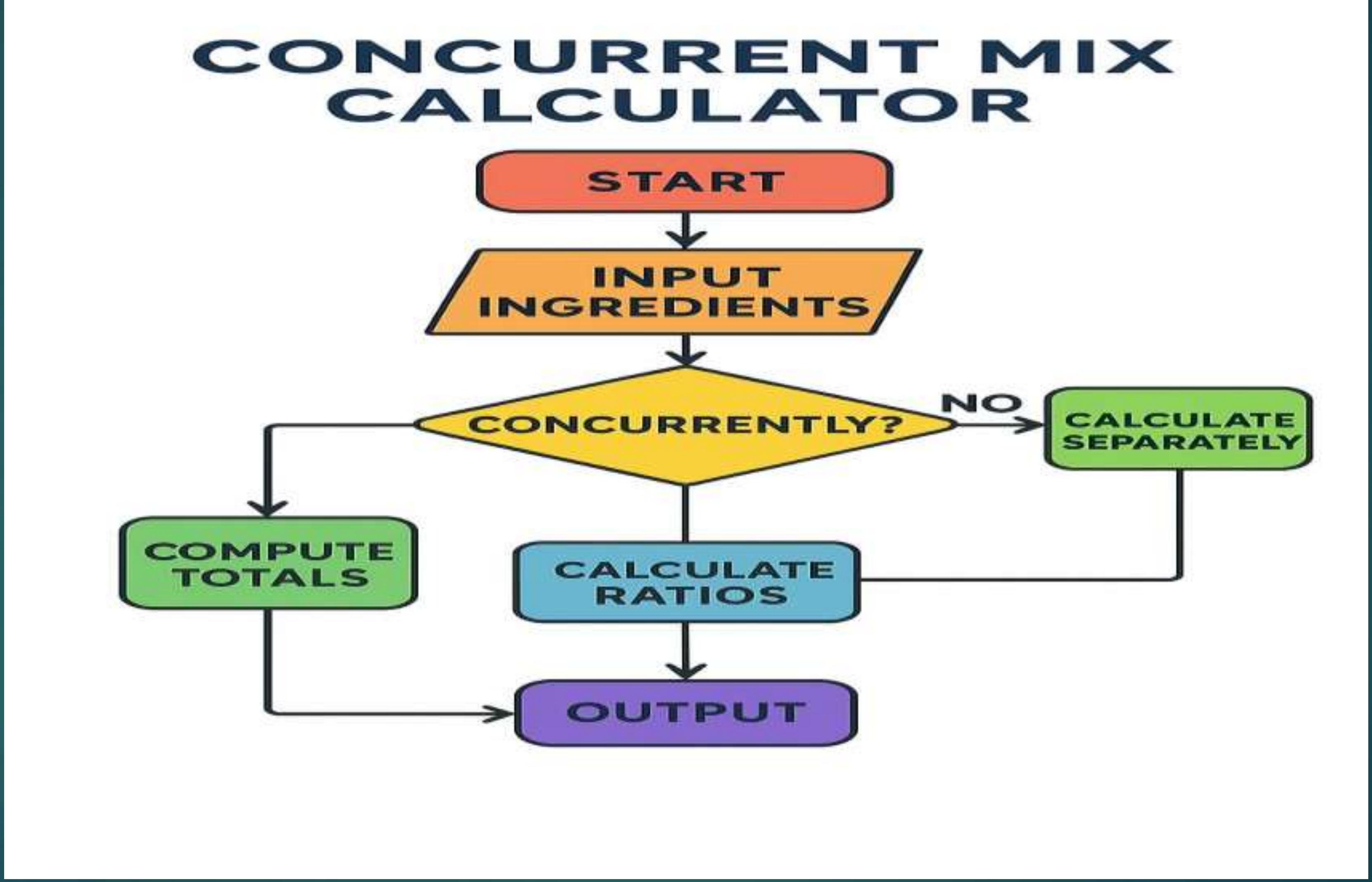
Algorithm

- ALGORITHM: Concurrent Mix Calculator and Beam Load Calculator
- ==
  PART A: CONCRETE MIX DESIGN CALCULATOR
- 1. INPUT MODULE
  Input Parameters:
- Grade of concrete (M15, M20, M25, M30, M35, M40, etc.)
- Type of cement (OPC 33, 43, 53)
- Maximum aggregate size (10mm, 20mm, 40mm)
- Degree of workability (slump value)
- Exposure conditions (mild, moderate, severe)
- Quality of materials (aggregates, water)
- 2. TARGET STRENGTH CALCULATION
  Process:
  a. Obtain characteristic strength (fck) from grade
  Example: M25 → fck = 25 N/mm²
- b. Calculate target mean strength (fm):
  fm = fck + 1.65 × s
  where s = standard deviation (depends on quality control)
- Good control: s = 4 N/mm²
- Fair control: s = 5 N/mm²
- Output: Target mean strength

- **3. WATER-CEMENT RATIO SELECTION**
  Process:
  a. Calculate W/C ratio from target strength:
  W/C ≈ 0.5 (for approximate calculation)
- Or use empirical formula:
  $fck = (fm \times Ce \times Wa) / (Ce + k \times Wa)$
  where Ce = cement content, Wa = water content, k = constant
- b. Check durability requirements:
- Mild exposure: max W/C = 0.55
- Moderate exposure: max W/C = 0.50
- Severe exposure: max W/C = 0.45
- c. Select minimum of strength and durability W/C ratio
- Output: Final W/C ratio
- **4. WATER CONTENT DETERMINATION**
  Process:
  a. Select water content based on:
  - Maximum aggregate size
  - Slump value required
- b. Use standard tables (IS 10262):
  For 20mm aggregate:
- 25–50mm slump: 186 liters
- 50–100mm slump: 205 liters
- Output: Water content per m^3
- **5. CEMENT CONTENT CALCULATION**
  Process:
  a. Calculate cement (C) from W/C ratio:
  C = Water content / (W/C ratio)
- b. Check minimum cement content (durability):
- Mild exposure: min 300 kg/m^3

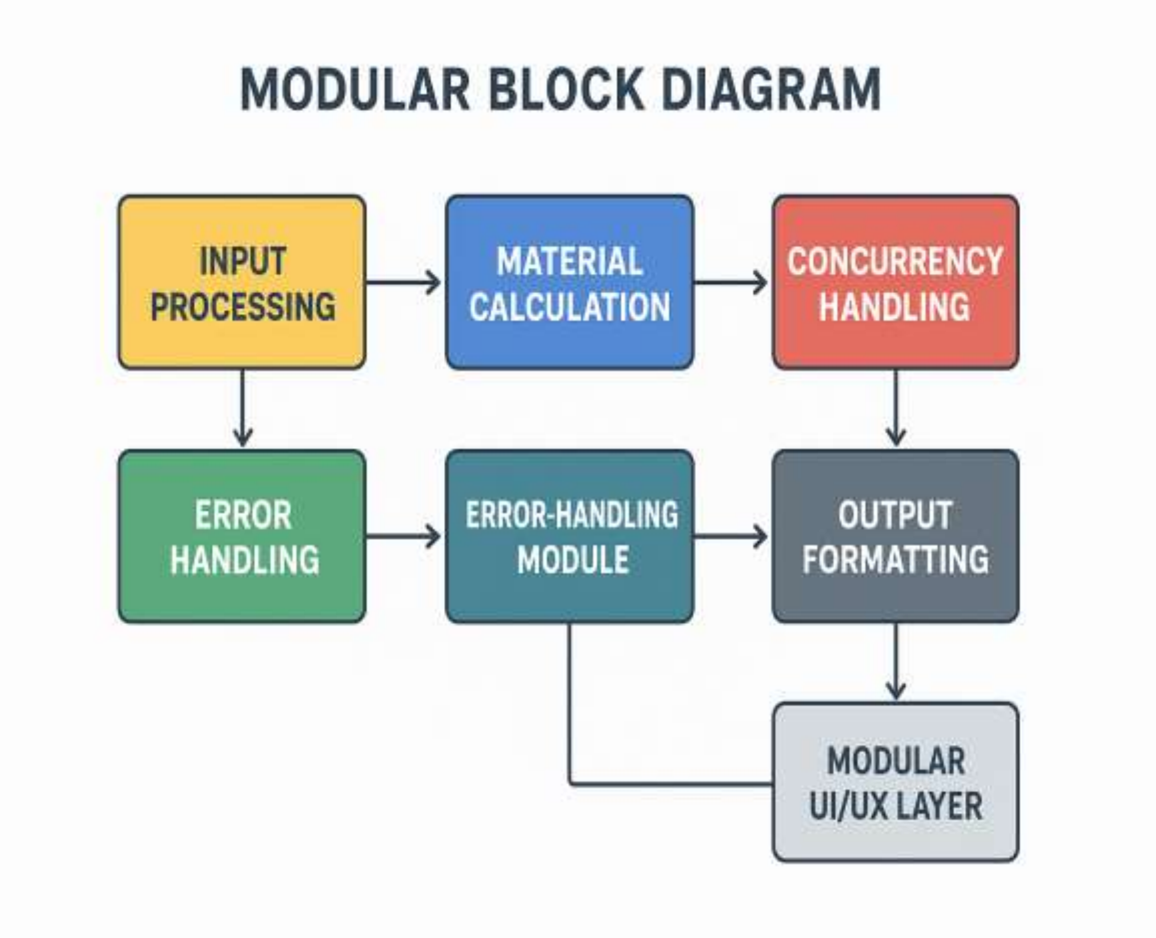
- Moderate exposure: min 320 kg/m³
- Severe exposure: min 340 kg/m³
- c. Use maximum of calculated and minimum values
- Output: Cement content (kg/m³)
- ## 6. AGGREGATE PROPORTIONING
  Process:
  a. Calculate total aggregate = 1 m³ – (Cement vol + Water vol)
- b. Determine fine aggregate (FA) to coarse aggregate (CA) ratio:
- Use grading zone and workability tables
- Typical ratio: 1:2 (FA:CA)
- c. Calculate individual quantities:
  FA = Total aggregate × FA ratio / (FA + CA ratio)
  CA = Total aggregate × CA ratio / (FA + CA ratio)
- Output: Fine aggregate and coarse aggregate quantities
- ## 7. MIX ADJUSTMENT FOR ADMIXTURES
  Process:
  a. If using admixtures (plasticizers, superplasticizers):
  - Reduce water content by 5–30%
  - Maintain W/C ratio
  - Adjust cement accordingly
- Output: Adjusted mix proportions
- ## 8. FINAL MIX PROPORTIONS
  Output Format:
- Cement : Fine Aggregate : Coarse Aggregate (by weight)
- Example: 1 : 1.5 : 3
- Quantities per m³:
  - Cement: XXX kg
  - Fine Aggregate: XXX kg
  - Coarse Aggregate: XXX kg
  - Water: XXX liters

Flowchart

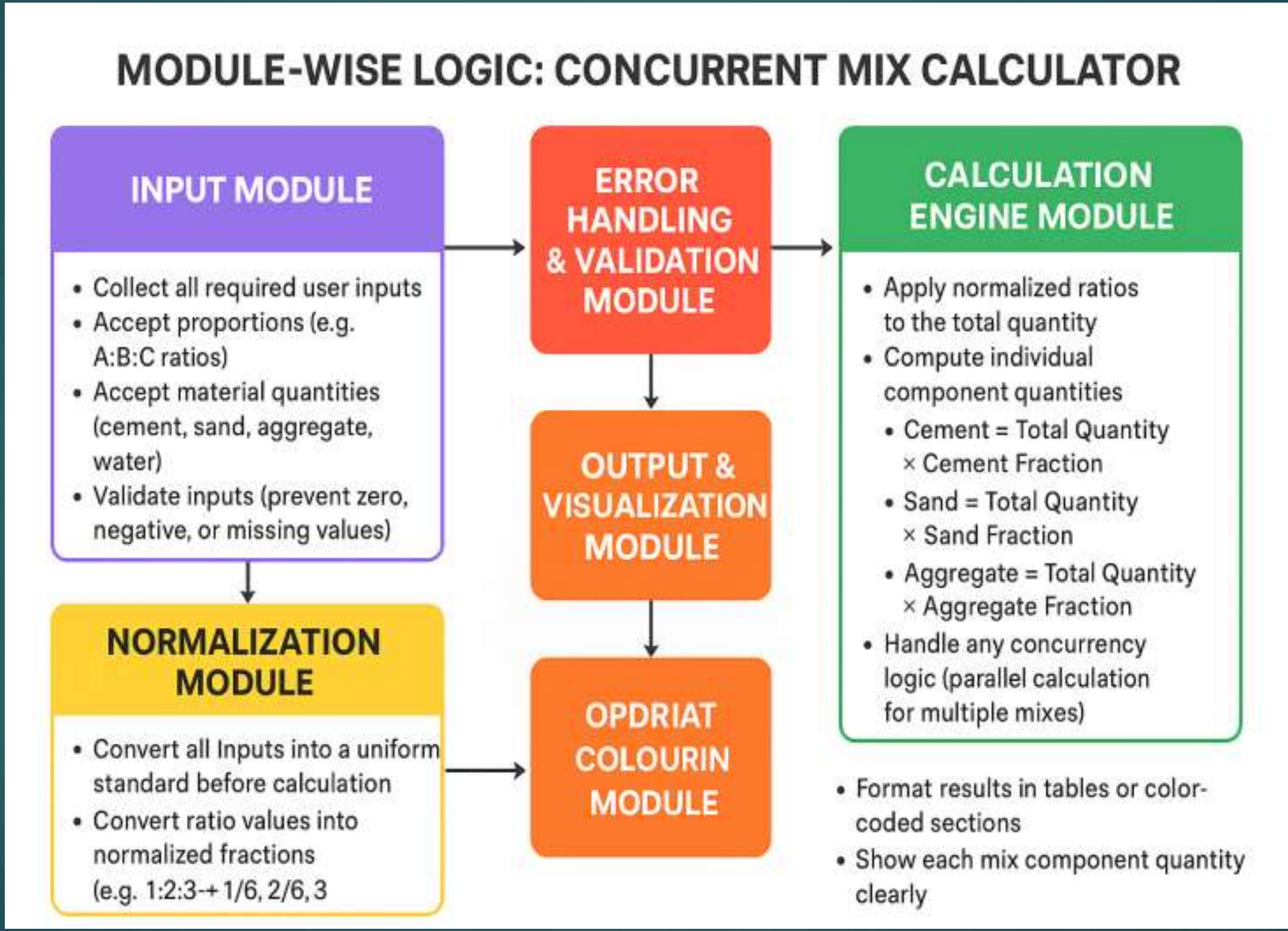

CONCURRENT MIX
CALCULATOR
START
INPUT
INGREDIENTS
CONCURRENTLY?
NO
CALCULATE
SEPARATELY
COMPUTE
TOTALS
CALCULATE
RATIOS
OUTPUT

Modular Design

- ❑ The system is divided into smaller, self contain units that can function independently
- ❑ Since modules are isolated, you can update or fix one module without effecting the entire system
- ❑ Modules can be reused in different projects or systems, reducing development time and cost
- ❑ New features or components can be added easily by plugging in new modules without designing the whole system
- ❑ Different team members can work on separate modules simultaneously, increasing efficiency and reducing dependencies



Module Wise Logic



Module Wise Program

```
#include<stdio.h>
Int main () {
float c, s, a, w, total;
    float cement, sand, aggregate, water;
    float sum;
    // Input
    printf("Enter cement ratio: ");
    scanf("%f", &c);
    printf("Enter sand ratio: ");
    scanf("%f", &s);
    printf("Enter aggregate ratio: ");
    scanf("%f", &a);
```

```
printf("Enter water ratio: ");
    scanf("%f", &w);
    printf("Enter total quantity (m³): ");
    scanf("%f", &total);
    // Calculation
    sum = c + s + a + w;
    cement = (c / sum) * total;
    sand = (s / sum) * total;
   aggregate = (a / sum) * total;
   water = (w / sum) * total;
```

```c
// Output
    printf("\n--- RESULTS ---\n");
    printf("Cement: %.2f units\n", cement);
    printf("Sand: %.2f units\n", sand);
    printf("Aggregate: %.2f units\n", aggregate);
    printf("Water: %.2f units\n", water);
    return 0;
```

Sample Output

- Enter cement ratio: 1
- Enter sand ratio: 2
- Enter aggregate ratio: 3
- Enter water ratio: 0.5
- Enter total quantity (m^3): 10

- // RESULTS
- Cement: 1.18 units
- Sand: 2.35 units
- Aggregate: 3.53 units
- Water: 0.59 units

THANK

YOU

BEAM LOAD CALCULATOR

REQUIREMENTS

HARDWARE REQUIREMENTS SOFTWARE REQUIREMENTS * Any computer or laptop * Windows 7/8/10/11

* Minimum 2 GB RAM * Linux

* Keyboard for input * Mac Os

* Personal computers * MS Word

INTRODUCTION :

A Beam Load Calculator is a program that calculates how forces act on a beam when loads are applied .

This helps students understand basic structural analysis concepts such as :

Reactions at supports

Shear force

Bending moment

The project teaches input handling , modular programming , and applying formulas in code .

ALGORITHM

Step - by - step algorithm

1. Start
2. Display menu of load types
3. Get beam length from user
4. Ask user to select load type :

o 1 – Point Load

o 2 – Uniformly Distributed Load (UDL)

5 . If Point Load :

o Get magnitude of load

o Get position of load

o Calculate reactions

o Calculate maximum bending moment

6 . If UDL :

o Get intensity (load per meter)

o Calculate reactions

- o Calculate maximum bending moment

7 . Display results

8 . End

FLOWCHART

Start___ calculateon of Sn(Eqs. 3 or 8)

|

Calculation of Lp assuming Sa=Sn (Eq. 2)

|

Calculation of Pdy (Eq. 1) ___ Calculation of Vcd (Eq. 7)

|

Vcd< Pdy

Scr= Sn

Scr= 2/3 Sn

Calculation of Lp (Eq. 2)

|

Lp< La

Total loss of FRP

Laminate’s efficiency

|

Pdy = Puc

yielding of longitudinal reinforcement in

substrate concret at the shear flexure

Zone

|

calculation of Pdy (Eq. 1)

|

Pdy = Pdy ___ Pdy < Vcd

Pdy = Vcd

MODULAR DESIGN

------------------------ +

MODULE |

--------------------------- +

+

| MAIN

+

|

```
+
----------------------------- +
| INPUT
MODULE |
+
-------------------------------- +
|
+
-------------------------------- +
|
VALIDATION MODULE |
+
------------------------------ +
|
+-------------------------------- +
|
CALCULATION MODULE |
+
-------------------------------- +
|
+
------------------------------- +
|
OUTPUT MODULE |
+
------------------------------- +
```

MODULE WISE LOGIC

MODULE WISE PROGRAM

```
#include <stdio.h>
Int input data(float *P ,float *W ,int *type) {
Printf("Beam Load Calculator\n");
Printf("1. Point Load at Center\n");
Printf("2. UDL (Uniformly Distributed Load)\n");
Printf("Enter Load Type:");
Scanf("% d",type);
Printf("Enter Length of Beam (in meters):");
scanf("% f" ,L);
If(*type==1) {
```

```
Printf("Enter Point Load (in KN):");
Scanf("% f" ,P);
} else {
printf("Enter UDL Load (KN/m):");
scanf("% f" ,W) ;
  }
}
Float calculate shear force(float L, float P, float W, int
type) {
If(type==1) {
Return P/2;
```

```
} else {
Return (W*L)/2;
  }
}
Float calculate bending moment (float L, float P, float W,
int type) {
      If(type==1) {
Return(P*L)/4;
} else {
Return (L*P)/8;
   }
```

```
}
 // OUTPUT
Int display results(float SF , float BM)  {
   Printf("\n-----Results----\n");
  printf("Shear force  =  % .2f kN\n",SF);
  printf("Bending moment =% .2f Kn- m\n",BM);
 printf("-----------------\n");
}
```

// SAMPLE OUTPUT

Point Load at Center

UDL (Uniformly Distributed Load)

```
Enter Length of Beam (in meters): 4
Enter Load Type : 1
Enter Point Load (in kN): 18

// RESULTS
Shear Force = 9.00 kN
Bending Moment = 18.00 Kn - m
```

THANK YOU