Factor of Safety Calculation

Objective

The purpose of this program is to compute the **Factor of Safety (FOS)** of a coal mine pillar based on various empirical strength formulas and stress estimation using the **Tributary Area Theory**. The FOS helps determine the safety level of the underground structure.

Programming Language

The program is written in **Python**, a widely used high-level programming language known for its readability and simplicity.

Functionality Overview

1. Stress Calculation

Based on the **Tributary Area Theory**, the vertical stress acting on a pillar is calculated considering:

- Overburden density (in g/cm^3)
- Depth of the coal seam (in meters)
- Gravitational acceleration (9.81 m/s^2)
- Pillar width and gallery width (in meters)

2. Strength Estimation

The user can choose from one of the following empirical formulas to calculate the **pillar strength**:

- **Obert-Duvall Formula:**

$$\text{Strength} = UCS \times (0.778 + 0.222 \times (w/h))$$

- **Holland-Gaddy Formula:**

$$\text{Strength} = k \times \sqrt{w/h}$$

- **Salamon Formula:**

$$\text{Strength} = 1320 \times \frac{w^{0.46}}{h^{0.66}}$$

- **Bieniawski Formula:**

$$\text{Strength} = \sigma_1 \times (0.64 + 0.36 \times (w/h))$$

3. FOS Calculation

$$\text{FOS} = \frac{\text{Strength}}{\text{Stress}}$$

4. Safety Remark

- $\text{FOS} \geq 1.8$ **Safe**
- $1.3 \leq \text{FOS} < 1.8$ **Moderately Safe**
- $\text{FOS} < 1.3$ **Unsafe**

User Inputs

- **For Stress Calculation:**
  - Average density of overlying strata (g/cm^3)
  - Depth of coal seam (m)
  - Pillar width (m)
  - Gallery width (m)
- **For Strength Calculation:**
  - Pillar height (m)
  - Depending on the selected method:
    * UCS (for Obert-Duvall)
    * k factor (for Holland-Gaddy)
    * Cubical coal strength (for Bieniawski)

Outputs

- Pillar Stress (MPa)
- Pillar Strength (MPa)
- Factor of Safety (FOS)
- Safety Remark (Safe / Moderately Safe / Unsafe)

Strengths of the Program

- **Modular Design:** Functions are well-separated by functionality.
- **User-Friendly:** Prompts guide the user clearly through input.
- **Comprehensive:** Includes multiple empirical strength estimation methods.
- **Safety Feedback:** Provides interpretation of FOS results.

Suggestions for Improvement

- Add **exception handling** to manage invalid inputs.
- Build a **graphical user interface (GUI)** using Tkinter or PyQt.
- Add **charts or visualizations** for better result presentation.
- Support **additional stress estimation models**.

Conclusion

This program is an effective tool for estimating the Factor of Safety for coal mine pillars. It integrates real-world engineering formulas into a computational model, making it useful for mining engineering students and professionals to assess pillar stability quickly and accurately.

Appendix: Python Code

```python
import math

def tributary_area_stress(density, gravity, depth, wp, wg):
    density = density * 10**3  # Convert g/cm  to kg/m
    return (density * gravity * depth * (wp ** 2)) / ((wp - wg) ** 2)

def obert_duvall_strength(c1, w, h):
    c1 = c1 * 10**6  # Convert MPa to Pa
    return c1 * (0.778 + 0.222 * (w / h))

def holland_gaddy_strength(k, w, h):
    return k * math.sqrt(w / h)

def salamon_strength(w, h):
    return 1320 * (w ** 0.46) / (h ** 0.66)

def bieniawski_strength(sigma1, w, h):
    sigma1 = sigma1 * 10**6  # Convert MPa to Pa
    return sigma1 * (0.64 + 0.36 * (w / h))

def factor_of_safety(stress, strength):
    return strength / stress

def remark(fos):
    if fos >= 1.8:
        return "Safe"
    elif 1.3 <= fos < 1.8:
        return "Moderately Safe"
    else:
        return "Unsafe"

def main():
    print("Select the method to calculate pillar stress:")
    print("1. Tributary Area Theory")
    stress_method = int(input("Enter the method number: "))

    if stress_method == 1:
        density = float(input("Enter average density of overlying strata (g/cm^3): "))
        gravity = 9.81  # m/s^2
        depth = float(input("Enter depth of the coal seam (m): "))
        pillar_width = float(input("Enter pillar width (m): "))
        gallery_width = float(input("Enter gallery width (m): "))

        wp = pillar_width + gallery_width  # Corrected center-to-center pillar width
        wg = gallery_width

        stress = tributary_area_stress(density, gravity, depth, wp, wg)
    else:
        print("Invalid selection!")
        return

    print("Select the method to calculate pillar strength:")
    print("1. Obert-Duvall Formula: UCS * (0.778 + 0.222 * (w / h))")
    print("2. Holland-Gaddy Formula: k * sqrt(w / h)")
```

```python
    print("3. Salamon Formula: 1320 * (w^0.46) / (h^0.66)")
    print("4. Bieniawski Formula: (cubical coal strength) * (0.64 + 0.36 * (w / h))")
    strength_method = int(input("Enter the method number: "))

    h = float(input("Enter pillar height (m): "))

    if strength_method == 1:
        c1 = float(input("Enter uniaxial compressive strength (MPa): "))
        strength = obert_duvall_strength(c1, wp, h)
    elif strength_method == 2:
        k = float(input("Enter K factor from lab tests: "))
        strength = holland_gaddy_strength(k, wp, h)
    elif strength_method == 3:
        strength = salamon_strength(wp, h)
    elif strength_method == 4:
        sigma1 = float(input("Enter strength of cubical coal specimen (MPa): "))
        strength = bieniawski_strength(sigma1, wp, h)
    else:
        print("Invalid selection!")
        return

    fos = factor_of_safety(stress, strength)
    fos_remark = remark(fos)

    stress = stress / 10**6  # Convert Pa to MPa
    strength = strength / 10**6  # Convert Pa to MPa

    print(f"\nPillar Stress: {stress:.2f} MPa")
    print(f"Pillar Strength: {strength:.2f} MPa")
    print(f"Factor of Safety: {fos:.2f}")
    print(f"Remark: {fos_remark}")

if __name__ == "__main__":
    main()
```