In [ ]:

```python
import numpy as np

class DragComponent:
    def __init__(self, C_f=None, FF=None, Q=None, S_wet=None):
        self.C_f = C_f
        self.FF = FF
        self.Q = Q
        self.S_wet = S_wet

    def CalculateDrag(self):
        return self.C_f * self.FF * self.Q * self.S_wet


def dragDragComponents(M, rho, V, mu):
    wing = DragComponent()
    hTail = DragComponent()
    vTail = DragComponent()
    nacelle = DragComponent()
    fuselage = DragComponent()

    wing.Q = 1
    hTail.Q = 1.04
    vTail.Q = 1.04
    nacelle.Q = 1.3
    fuselage.Q = 1

    wing.S_wet = 1678.729       #ft^2
    hTail.S_wet = 186.192
    vTail.S_wet = 286.139
    nacelle.S_wet = 127.751
    fuselage.S_wet = 1830.348
    # Reference Lengths for Cf
    fuselage_L = 79      #ft
    nacelle_L = 18.543
    wing_L = 7.15871
    h_tail_L = 4.59891
    v_tail_L = 11.3088

    ReW = (rho * V * wing_L) / mu
    ReH = (rho * V * h_tail_L) / mu
    ReV = (rho * V * v_tail_L) / mu
    ReN = (rho * V * nacelle_L) / mu
    ReF = (rho * V * fuselage_L) / mu


    # Max thickness of wings and tails (ratio)
    tc_wing = 0.096093
    tc_h_tail = 0.119119
    tc_v_tail = 0.119119


    # Chordwise location of max thickness (ratio)
    xc_wing = 0.35749
    xc_h_tail = 0.298547
    xc_v_tail = 0.298547

    # Sweep angle of wings and tails IN RADIANS
    gammaW = 0.087266       #rad
    gammaH = 0.279253
    gammaV = 0.715243

    # Fuselage form factor
    Amax =  86.1771 # Maximum cross sectional area of fuselage
    f = fuselage_L / (np.sqrt((4/np.pi)*Amax)) # Fineness Ratio
    if f < 6:
        fuselage.FF = (0.9 + (5 / (f**1.5)) + (f/400))
    else:
        fuselage.FF = (1 + (60 / (f**3)) + (f/400))


    # Nacelle form factor
    nacelle.FF = (1 + (0.35/f))
```

```python
    # Wings and tails
    wing.FF = (1 + (0.6 / xc_wing)*tc_wing + 100*(tc_wing**4)) * (1.34 * (M**0.18)*(np.cos(gammaW)**0.28))
    vTail.FF = (1 + (0.6 / xc_v_tail)*tc_v_tail + 100*(tc_v_tail**4)) * (1.34 * (M**0.18)*(np.cos(gammaV)**0
    hTail.FF = (1 + (0.6 / xc_h_tail)*tc_h_tail + 100*(tc_h_tail**4)) * (1.34 * (M**0.18)*(np.cos(gammaH)**0

    # print('formfactor',wing.FF,vTail.FF,hTail.FF)


    def Cfcalc(Re, wL, wT):
        Cf_lam = 1.328 / np.sqrt(Re)
        Cf_turb = 0.455 / (((np.log10(Re))**2.58) * ((1 + 0.144*(M**2))**0.65))
        Cf = (Cf_lam * wL) + (Cf_turb * wT)
        return Cf

    wing.C_f = Cfcalc(ReW, 0.5, 0.5)
    hTail.C_f = Cfcalc(ReH, 0.5, 0.5)
    vTail.C_f = Cfcalc(ReV, 0.5, 0.5)
    fuselage.C_f = Cfcalc(ReF, 0.25, 0.75)
    nacelle.C_f = Cfcalc(ReN, 0.1, 0.9)

    return [wing, hTail, vTail, fuselage, nacelle]
```